

A CLAIMEANGUAGES	WHAT AS CLAYMOF INFRINGEMENT AND A
155.	Products infringing: Any product using Microsoft Product Activation or Reader Activation feature.
A virtual distribution environment comprising	
(a) a first host processing environment comprising	computer running a Microsoft product containing the Product Activation feature, including Windows XP, Office XP, Visio
	2002. Reader using its activation feature.
(1) a central processing unit;	CPU of computer
(2) main memory operatively connected to said central processing unit;	main memory of computer
(3) mass storage operatively connected to said central processing unit and said main memory;	hard disk or other mass storage contained in computer
(b) said mass storage storing tamper resistant software designed to be loaded into said main	Microsoft Product Activation software
memory and executed by said central processing unit, said tamper resistant software	
comprising:	Product Activation software generates
(1) machine check programming which derives information from one or more	hardware information relating to the host
aspects of said host processing environment,	processing environment as part of the activation process
(2) one or more storage locations storing said information;	hardware information is stored in the computer's storage
(3) integrity programming which	
(i) causes said machine check programming to derive said	each time the Microsoft program starts up after initial activation, Product Activation checks
information,	the originally derived hardware information against current hardware
(ii) compares said information to information previously stored	each time the Microsoft program starts up after initial activation, Product Activation checks
in said one or more storage locations, and	the originally derived hardware information against current hardware
(iii) generates an indication based on the result of said	Product Activation software indicates whethe the test has passed or failed
comparison; and	
(4) programming which takes one or more actions based on the state of said indication;	
(i) said one or more actions including at least temporarily	Product Activation software will allow system startup procedures to continue, if test succeed
halting further processing.	or discontinue startup and offer user opportunity to reactivate if the test fails
	1 Opportunity to reach vale is the test rang

Exhibit B

293482.02

27

•	<b> </b>	
5	156.	Product Infringing: Any product using Microsoft Product Activation or Reader Activation feature.
_	A virtual distribution environment comprising	
6	(a) a first host processing environment	computer running a Microsoft product
7	comprising	containing the Product Activation feature, including Windows XP, Office XP, Visio 2002 and Reader
8		
	(1) a central processing unit;	CPU of computer
9	(2) main memory operatively connected to said central processing unit;	main memory of computer
10	(3) mass storage operatively connected to said central processing unit and said main memory;	hard disk or other mass storage contained in computer
11	(b) said mass storage storing tamper resistant	Microsoft Product Activation software
12	software designed to be loaded into said main memory and executed by said central	171010101111000011100110011001100110011
13	processing unit, said tamper resistant software comprising:	
14	(1) machine check programming which derives information from one or more	Product Activation software generates hardware information relating to the host
15	aspects of said host processing environment,	processing environment as part of the activation process
16	(2) one or more storage locations storing said information;	hardware information is stored in the computer's storage
17	(3) integrity programming which	
18	(i) causes said machine check programming to derive said information,	each time the Microsoft program starts up after initial activation, Product Activation checks the originally derived hardware information against current hardware
19	(i) commerce said information	each time the Microsoft program starts up after
20	(ii) compares said information to information previously stored in said one or more storage	initial activation, Product Activation checks the originally derived hardware information
21	locations, and	against current hardware
22	(iii) generates an indication based on the result of said	Product Activation software indicates whether the test has passed or failed
	comparison; and	
23	(4) programming which takes one or more actions based on the state of said	
24	indication;	
25	(i) said one or more actions including at least temporarily	Product Activation may disable the underlying software from generating new files or running
26	disabling certain functions.	user applications if the test fails
- 11		<i>,</i>

Exhibit B

27

293482.02

4		
5	157.	Product Infringing: Any product using Microsoft Product Activation or Reader Activation feature.
6	A virtual distribution environment comprising	Med variou leature.
U	(a) a first host processing environment	computer running a Microsoft product
7	comprising	containing the Product Activation feature, including Windows XP, Office XP, Visio 2002
.8		and Reader
. 9:	(1) a central processing unit;	CPU of computer
10	(2) main memory operatively connected to said central processing unit;	main memory of computer
. 10	(3) mass storage operatively connected	hard disk or other mass storage contained in
- 11	to said central processing unit and said main memory;	computer
12	(b) said mass storage storing tamper resistant software designed to be loaded into said	Microsoft Product Activation software
13	main memory and executed by said central processing unit, said tamper resistant	
1.4	software comprising: (1) machine check programming which	Product Activation software generates hash
15	derives information from one or more aspects of said host processing	information relating to the host processing environment as part of the activation process
16	environment,	
	(2) one or more storage locations	hardware information is stored in the
17	storing said information; (3) integrity programming which	computer's storage
10	(i) causes said machine check	anch time the Minnesses
18	programming to derive said	each time the Microsoft program starts up after initial activation, Product Activation checks
19	information,	the originally derived hardware information against current hardware
20	(ii) compares said information to information previously stored	each time the Microsoft program starts up after initial activation, Product Activation checks
21	in said one or more storage locations, and	the originally derived hardware information against current hardware
22	(iii) generates an indication based on the result of said	Product Activation software indicates whether the test has passed or failed
23	comparison; and	
24	(4) programming which takes one or more actions based on the state of said indication;	
25	(i) said one or more actions including displaying a message	Product Activation software displays a message to the user if the test fails
26	to the user.	
1		•

_ 3	SELECTION OF THE PROPERTY OF T	THE REPORT OF THE PROPERTY OF
5	156.	Products infringing: Windows Media Player
- 11	A virtual distribution environment comprising	
	a first host processing environment comprising	WMP with Individualized DRM client
- 41	a mot not processing on mountain	(referred to hereafter as the Individualized
7		WMP) running on a client computer
. H.	a central processing unit	Client CPU
8	main memory operatively connected to said	Client memory
	central processing unit	
	mass storage operatively connected to said	Local disk drive
- 11	central processing unit and said main memory	Boom disk cirve
	said mass storage storing tamper resistant	Individualized WMP (I-WMP) stored on disk
31	said mass storage storing tamper resistant software designed to be loaded into said main	and loaded into main memory upon execution
	memory and executed by said central	I-WMP is tamper resistant.
	processing unit, said tamper resistant software	1- With is tamper resistant.
!   F	comprising:	
, H <u>S</u>	machine check programming which derives	Individualization module is generated by the
Ţ	information from one or more aspects of said	MS individualization service either when the
	host processing environment,	un-individualized WMP tries to open licensed
1	lost processing environment,	content that requires a security upgrade (aka,
		Individualization) or when the user requests a
I		upgrade un-provoked. The individualization
.		module is unique and signed and is bound to
		unique hardware ID using the MS machine
		activation process.
1	one or more storage locations storing said	The aforementioned unique feature are locate
i	nformation	in multiple places or storage locations
	ntegrity programming which	In manage places of closings recovering
	causes said machine check programming to	The ID is regenerated by WMP/DRM client
	lerive said information,	when first loading the Individualized DRM
,    °	icitye said illiotiiladoli,	Client to access a piece of content requiring t
		security upgrade.
1	compares said information to information	The program checks the new copy against the
	previously stored in said one or more storage	one to which the Individualized DRM client i
	ocations, and	bound.
	penerates an indication based on the result of	Program stores the result of this check.
		1 togram stores the result of this cheek.
S	aid comparison; and	If these are not equal, the user is notified via
p b	programming which takes one or more actions	message stating that he/she must acquire a
b	pased on the state of said indication	incode Stating that is the overest consists
		security upgrade (that is, the current security
H		upgrade is invalid). If they are equal then
.		processing of songs requiring Individualization
		continues.
S	aid one or more actions including at least	Songs targeted to this Individualization modu
1 16	emporarily disabling certain functions.	cannot be accessed until the upgrade is correct

Exhibit B

293482.02

2,8

1

2

157. A virtual distribution environment comprising	Infringing products include: Windows Media Player
a first host processing environment comprising	See 156
a central processing unit	See 156
main memory operatively connected to said central processing unit	See 156
mass storage operatively connected to said central processing unit and said main memory	See 156
said mass storage storing tamper resistant software designed to be loaded into said main	See 156
memory and executed by said central processing unit, said tamper resistant software	
comprising:	
machine check programming which derives information from one or more aspects of said host processing environment,	See 156
one or more storage locations storing said information	See 156
integrity programming which causes said machine check programming to derive said	See 156
information compares said information to information previously stored in said one or more storage locations, and	
generates an indication based on the result of	See 156
said comparison; and	DC 130
programming which takes one or more actions based on the state of said indication	See 156
said one or more actions including displaying a message to the user.	If these are not equal, the user is notified via a message stating that he/she must acquire a security upgrade (that is, the current security upgrade is invalid).

.22

Exhibit B
5

293482.02

,		District Control of the Control of t
. 4	WEST SECTION OF THE PROPERTY O	CEATMORIN FRINGEMENTS
•	157.	Infringing Product: Microsoft's Windows File
. 5		Protection and System File Checker features, embodied in Microsoft's Windows 2000,
		Windows XP products, and Server 2003
6	A virtual distribution environment comprising	Willdows AT products, and Server 2003
	(a) a first host processing environment	computer running Microsoft Windows 2000 or
7	comprising	Windows XP.
8	comprising	
9	(1) a central processing unit;	CPU of computer
	(2) main memory operatively connected	main memory of computer
10	to said central processing unit;	
11	(3) mass storage operatively connected	hard disk or other mass storage contained in
11	to said central processing unit and said	computer
12	main memory;	Windows File Protection process/service
	(b) said mass storage storing tamper resistant software designed to be loaded into said	("WFP") and System File Checker (SFC.exe)
13	main memory and executed by said central	features of winlogon.exe. Winlogon.exe is
14	processing unit, said tamper resistant	treated as a "critical" service by the Windows
14	software comprising:	operating system. Files supporting WFP
15		(including winlogon.exe, sfc.exe, sfc.dll (2000
		only), sfcfiles.dll (2000 only) and sfc_os.dll (XP only)) are "protected" files and are signed
16		using a signature verified by a hidden key. In
17	·	Windows 2000, WFP uses hidden functions
17		within the sfc.dll library. Functions are
18		imported by "ordinal" instead of "name."
	(1) machine check programming which	Winlogon either directly or using another dll (XP) or using SFC.dll (2000) determines if
19	derives information from one or more aspects of said host processing	changed file was protected, computes the hash
	environment,	of protected files and, if necessary, computes
20	Christinon,	the hash of the file in the dll cache before using
21		it to replace a file overwritten by an incorrect
		version of the file.
22	(2) one or more storage locations	hardware information is stored in the
	storing said information;	computer's memory
23	(3) integrity programming which (i) causes said machine check	Windows notifies Winlogon when there has
	programming to derive said	been a system directory change or a change in
24	information,	the dll cache.
25		
26	(ii) compares said information	Winlogon either directly or using another dll
	to information previously stored	(XP) or using SFC.dll (2000) compares
27	in said one or more storage	computed hash with hash in the hash database created from the Catalog file(s), and, if there is
26	locations, and	a difference, compares the hash of the file in
28		the dll cache to the hash database created from
1		
	,	
1	·	1 4 11 11

Exhibit B 6

1

2

•	(iii) generates an indication	overwritten file.
•	based on the result of said comparison; and	An event is written to the Event Viewer if hashes do not agree.
	(4) programming which takes one or more actions based on the state of said indication;	Depending on the circumstances, WFP displays several messages to the user, including prompting the user to contact the system administrator, and to insert a CD-ROM
	(i) said one or more actions including displaying a message to the user.	See above. Messages also constitute viewable Event Property pop-ups.
•	· ·	
•		
	-	
	•	
		•
	• • • • • • • • • • • • • • • • • • • •	
		<del>.</del>
		•

4	**************************************	CENTROPINERINGEMENTS
		Product Infringing: XBox
5 6	A process comprising the following steps:	The process constitutes assembly and use of components making up an XBox game.
7	accessing a first record containing information directly or indirectly	The first record consists of the second file table on an XBox DVD. This table
.8	identifying one or more elements of a first component assembly,	identifies the .xbe file which includes the game information.
. 9		
10	at least one of said elements including at least some executable programming,	The xbe file includes executable programming.
11		
12	at least one of said elements constituting a load module.	The xbe file is a load module.
13	said load module including executable programming and a header;	The xbe file includes a header.
14	at least a portion of said header is a public portion which is characterized by a relatively lower level of security	Most information the xbe header is not obfuscated.
15	protection; and at least a portion of said header is a private	The entry point address and the kernel
16 17	portion which is characterized, at least some of the time, by a level of security protection which is relatively higher than said relatively lower level of security	image thunk address listed in the xbe header are obfuscated and therefore at a higher level of security protection.
18 19	protection, using said information to identify and locate said one or more elements;	The second file table identifies the .xbe file, including where that file is located.
. 20	accessing said located one or more elements;	The .xbe file is accessed by the XBox.
21	securely assembling said one or more elements to form at least a portion of said	At runtime, the .xbe file is assembled with
22	first component assembly;	certain services of the operating system to form a component assembly. Security
23		associated with this assembling process includes verifying signatures associated
24		with portions of the .xbe file, and replacing obfuscated calls to operating system
25		services with actual addresses.
26		The assembly may also include patch files downloaded from a remote server.
27		
28		Const. 1
ł	executing at least some of said executable	Game play requires execution of the

Exhibit B

1	programming; and	assembled programming.
. 3	checking said record for validity prior to	The second file table is protected by a
; <b>2</b>	performing said executing step.	digital signature, and is not loaded/used
3		unless the digital signature is verified
_		against the file.
4		
5	7. A process as in claim 6 in which:	
	said relatively lower level of security	The header is protected by the techniques
6	protection comprises storing said public header portion in an unencrypted state; and	protecting the xbe such as signing and security descriptors, but it is not encrypted
7	licader portion in an alienery processance, and	except as noted below.
•	said relatively higher level of security	The entry point address and the kernel
8	protection comprises storing said private	image thunk address listed in the xbe
` <b>.</b>	header portion in an encrypted state.	header are obfuscated. The Xbox SDK's (XDK) image build uses a key value shared
9		with the retail XBox to perform two XOR
10		operations against the addresses
•		
11		
12		
12		
13		
14		
14	•	
15		
16		
17		
		·
18	•	
19		
		•
20		·
	· ·	
21	•	
22		

. 

3	FOR U.S. PATENT NO. 5,917,912	
4	STATE OF ANY LANGUAGE WAS ASSESSED.	EXECUTOR THE RESTRICT OF THE PROPERTY OF THE P
5	8.	Infringing products: Microsoft CLR or CCLR and .NET Framework SDK and products that include one or both of these.
6	A comparison the following stone:	
,	A process comprising the following steps:  (a) accessing a first record containing	The first record is either an assembly manifest,
8	information directly or indirectly identifying one or more elements of a first component assembly,	or a whole assembly; the elements are other assemblies that are referenced as external in the first record; the first component assembly
	assemoly,	is a .NET application domain.
,   	(1) at least one of said elements including at least some executable programming,	Assembly contains executable programming.
	(2) at least one of said elements constituting a load module,	This is an external assembly referenced in the first record.
2	(i) said load module including executable programming and a header;	Assemblies include executable programming, and the assembly manifest and CLS type metadata constitute a header.
	(ii) said header including an	This feature is provided for in the .NET
5	execution space identifier identifying at least one aspect of	architecture through numerous mechanisms, for example, by demands for ZoneID
6	an execution space required for use and/or execution of the load module associated with said	permissions.
7	header;	
3	(iii) said execution space identifier provides the capability for distinguishing between	SecurityZone or other evidence provides this capability.
	execution spaces providing a higher level of security and execution spaces providing a	
	lower level of security;	
	(b) using said information to identify and locate said one or more elements;	Manifest and type metadata information section is used to identify and locate files, code elements, resource elements, individual classes and methods.
	(c) accessing said located one or more elements;	Step carried out by the CLR or CCLR loader.
	(d) securely assembling said one or more elements to form at least a portion of said first component assembly;	CLR or CCLR carries out this step, including checking the integrity of the load module, checking the load module's permissions, placing the load module contents into an application domain, isolating it from malicious or badly behaved code, and from code that does not have the permission to call it.
	(e) executing at least some of said executable programming; and	Step carried out by the CLR/CCLR and the CLR/CCLR host.

Exhibit B

1	(f) checking said record for validity prior to	The CLR/CCLR checks the authenticity and
2	performing said executing step.  9. A process as in claim 8 in which said	the integrity of the first .NET assembly.  The CLR/CCLR constitutes a secure
3	execution space providing a higher level of	processing environment.
4	security comprises a secure processing environment.	
•	13. A process as in claim 8 further comprising: (a) comparing said execution space identifier	In one example, the
5	against information identifying the execution	ZoneIdentityPermissionAttribute SecurityZone value demanded by control in the assembly
. 6	space in which said executing step is to occur; and	manifest is compared against the SecurityZone
7		attribute value corresponding to the calling method
8	(b) taking an action if said execution space identifier requires an execution space with a	CLR/CCLR will throw an exception and transfer control to an exception handler in the
. 9	security level higher than that of the execution	calling routine, or it will shut down the application if there is no such exception
10	space in which said executing step is to occur.	handler, if the permissions do not include the
11		permissions required by the ZoneIdentityPermissionAttribute. The
. 12		ZoneIdentityPermissions are hierarchical, unless customized.
13	14. A process as in claim 13 in which said action includes terminating said process prior	CLR/CCLR may terminate the process or transfer control to an exception handler that
	to said executing step.	may itself terminate the process.
. 14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
.		
25		
26		·
27		
28		r programme to the second sec
		#

٠		
4	THE TOTAL METALLINGUAGE.	GEATMOHINIRINGEMENTS
5	8.	Products infringing include Windows Installer SDK, and products that include the Windows
6		Installer technology.
7	A process comprising the following steps:	Scenario 1: use of Windows Installer packages (i.e. MSI files) to create Windows Installer-enabled applications, such as Office 2000 and
8		used of the WI service to install them.  Scenario 2: software distribution technologies
9		that use the Windows Installer OS service for installation, such as Internet Component
10		Download and products like Office Web Components.
11		Either scenario can be used by SMS, IntelliMirror and third party tools like
12		InstallShield and WISE. NT or later operating systems (because they
13		use the subsystem identifier) using cabinet files, .CAB, (because they have a
14		manifest and INF and/or OSD files), and have been signed with a digital signature and
15		will be authenticated by Authenticode or WinVerifyTrust API and
16		contain at least one PE (portable executables)
17		S I S I S I S I S I S I S I S I S I S I
18	(a) accessing a first record containing information directly or indirectly identifying	Scenario 1: First record is the .MSI file that contains information on what goes in the assembly and how to install the assembly.
19	one or more elements of a first component assembly,	Scenario 2:
20		A. First record is the cabinet manifest (indirect instructions)
21		
22		B. Or, First record can be INF and/or OSD files (direct instructions)
23		
24	(1) at least one of said elements including at least some executable	Both scenarios: The PE (portable executable) in the cabinet file is the executable
25	programming,	programming.
25		Both scenarios: PE is a load module:
27	(2) at least one of said elements constituting a load module,	
28	(i) said load module including executable programming and a	Both scenarios: The PE has several headers.

Exhibit B

header;	
(ii) said header including a execution space identifier identifying at least one aspan execution space require use and/or execution of the module associated with sain header;	PE Optional Header that is an execution space ect of d for cload
(iii) said execution space identifier provides the capa for distinguishing between execution spaces providing higher level of security and execution spaces providing	and those that can run in user mode. This is a key security concept of process separation the was introduced with Windows NT.
lower level of security;	The Subsystem field in the PE header is used by the system to indicate whether the executable will run within Ring 3 (user mode)
	or use Ring 0 (native or kernel mode).  Anything running in Ring 3 is limited to its own processing space. Executables running Ring 0 can reach out to other spaces and hav
	security measure built around them.
(b) using said information to identify and locate said one or more elements;	Scenario 1: the MSI file identifies and locate the elements
	Scenario 2: .CAB manifest is used to identify Physical location
	OSD and/or INF is used to identify Logical location
(c) accessing said located one or more	Scenario 1: Using the MSI file
elements;	Scenario 2: Using INF and/or OSD in cabine
	file
(d) securely assembling said one or more elements to form at least a portion of said	Both scenarios: Using the Window Installer OS service with various properties and flags
component assembly;	the settings for higher protection.
	Windows Installer has numerous flags that the developer can set to indicate how the assembly the installed in what privilege level with
	will be installed, in what privilege level, with how much user interface, and how much abil the user has to watch or change what is
	occurring. These controls have been strengthened with each release of Windows
	Installer. Windows Installer 1.1 and later has the ability to limit the users capabilities during

environment and later, using the Group Policybased Change and Configuration Management, the administrator has the most control

Fields that can be set by the developer or administrator to control what users can do include the following:

Transformssecure can be set to a value of 1 to inform the installer that transforms are to be cached locally on the user's computer in a location the user does not have write access. (Transforms create custom installations from a basic generic installation, for example to make the Finance versions different from the Marketing version or English versions different from Japanese versions.)

AllowLockdownBrowse and DisableBrowse can prevent users from browsing to the sources.

SourceList can be used to specify the only allowable source to be used for the installation of a given component.

Environment can be used to specify whether the installation can be done while the user is logged on or only when no user is logged on.

Security Summary Property conveys whether a package can be opened as read-only or with no restriction.

Privileged Property is used by developers of installer packages to make the installation conditional upon system policy, the user being an administrator, or assignment by an administrator.

Restricted Public Properties can be set as variables for an installation. "For managed installations, the package author may need to limit which public properties are passed to the server side and can be changed by a user that is not a system administrator. Some are commonly necessary to maintain a secure environment when the installation requires the installer use elevated privileges."

SecureCustomProperties can be created by the author of an installation package to add controls beyond the default list.

MsiSetInternalUI specifies the level of user interface from none to full.

A Sequence Table can be used to specify the required order of execution for the installation process. There are three modes, one of which is the Administrative Installation that is used by the network administrator to assign and install applications.

InstallServicesAction registers a service for the system and it can only be used if the user is

•		
1 2 3 4 5 6		an administrator or has elevated privileges with permission to install services or that the application is part of a managed installation.  DisableMedia system policy disables media sources and disables browsing to media sources. It can be used with DisableBrowse to secure installations version 1.1 that doesn't have some of the other capabilities.  AlwaysInstallElevated can be set per user or per machine and is used to install managed applications with elevated privileges.
7 8 9 10		AllowLockdownBrowse, AllowLockdownMedia and AllowLockdownPatch set these capabilities so they can only be performed by an administrator during an elevated installation. [See article "HowTo: Configure Windows Installer for Maximum Security (Q247528).
11 12 13		Windows XP Professional and .NET have the additional capability to set Software Restriction Policies and have these used by Windows Installer.  In addition, most of the software distribution
14 15 16		technologies that use Windows Installer also add a layer of their own controls. For example, SMS 2.0 enables the administrators to control the installation is optional or required and whether the user can affect the installation contents/features at all.
17 18 19	(e) executing at least some of said executable programming; and	Both scenarios: Part of executable is called during installation in order to do self-registration or perform custom actions. The overall executable is used at runtime.
20 21	(f) checking said record for validity prior to performing said executing step.	Scenario 1: Sign the overall package and the cabinet files.
22		Scenario 2: The cabinet file is signed.  For IE with the default security level or higher,
23		the digital signature is verified by Authenticode or a similar utility before the component is allowed to be assembled.
24		Component is anowed to be assembled.
25 26		
27		
28		
	Exhi	ibit B

3	FOR U.S. PATE	NT NO. 5,917,912
5	35.	Products infringing include all products that host the Microsoft .NET Common Language Runtime or Compact Common Language Runtime.
6	A process comprising the following steps:	
7	(a) at a first processing environment receiving a first record from a second processing environment remote from said first processing	Computer running the Microsoft CLR/CCLR receives, for example, a shared assembly header or a complete shared assembly from
8	environment;	another computer, for example a server.
9	(1) said first record being received in a secure container;	The shared assembly is cryptographically hashed and signed.
10	(2) said first record containing identification information directly or	The first record is either an assembly manifest, or a whole assembly; the elements are other
11	indirectly identifying one or more elements of a first component	assemblies that are referenced as external in the first record; the first component assembly
12	assembly; (i) at least one of said elements	is a .NET application domain.  Assembly contains executable programming.
13	including at least some executable programming;	
14	(ii) said component assembly allowing access to or use of specified information;	The specified information can include any kind of data file, stream, log, environment variables, etc.
15	(3) said secure container also including a first of said elements;	The shared assembly includes at least some executable programming.
16 17	(b) accessing said first record	CLR/CCLR accesses the assembly or assembly header.
18	(c) using said identification information to identify and locate said one or more elements;	Manifest and type metadata information section is used to identify and locate files, code elements, resource elements, individual classes and methods.
19	(1) said locating step including locating	Met by a multifile assembly, with files
20	a second of said elements at a third processing environment located remotely from said first processing	distributed across a network, or by the second element constituting another referenced assembly located elsewhere; the CLR/CCLR
21 22	environment and said second processing environment;	uses probing to locate and access the file.
23	(d) accessing said located one or more	Step carried out by the CLR/CCLR loader.
24	elements; (1) said element accessing step	Step carried out by the CLR/CCLR loader.
25	including retrieving said second element from said third processing environment;	
26	(e) securely assembling said one or more elements to form at least a portion of said first	CLR/CCLR carries out this step, including checking the integrity of the load module,
27	component assembly specified by said first	checking the load module's permissions, placing the load module contents into an
28	record; and	application domain, isolating it from malicious or badly behaved code, and from code that
,		<b>1</b>

(f) executing at least some of said executable programming.	does not have the permission to call it.  Step carried out by the CLR/CCLR.
(1) said executing step taking place at said first processing environment.	CLR/CCLR is operating in the first processing environment specified above.
:	

·4		<u> </u>
5	34.	Product Infringing: Microsoft Operating Systems that support device driver signature technology
6 7	A descriptive data structure embodied on a computer-readable medium or other logic	Signature technology
8	device including the following elements:  a representation of the format of data contained in a first rights management data	The driver package's INF is a data structure. The INF contains multiple types
9.	structure	of sections, structured as hierarchy /"branches," that the Windows operating
10		system or its Plug and Play and/or Set-up installation services "branch" through based on the operating system information
11		and device for which a driver is to be installed. The installation services use the
12 13		"branching" structure (format) to determine what files should be installed. The INF,
14		further provides disk location information and file directory path information for the files identified as necessary as a result of
15		the "branching" process.
16		The driver package is a "rights management" data structure based on the fact that it is governed and based on the
17		fact that it processes governed information.
18 19		Rights Management as Governed Item
20		A driver manufacturer can include rules governing the driver's installation and/or use in the driver's INF file. For example:
21		Security entries specify an access control list for the driver.
22		Driver developers can specify rules that determine behavior of the driver package
23 24		based on the user's operating system version, including product type and suite
25		and the device for which the driver is to be installed Rules specifying logging
26		Local administrators can establish policy as
27 <b>28</b>		to what action or notification should occur in the event that a driver being installed is not signed.
~~		L

1

2

	<b>!</b>	
1 2 3 4 5	**	The operating system installation services have a ranking criteria it follows when multiple drivers are available for a newly detected device. The criterion is used to determine the driver best suited for ensuring compatibility with the operating system and ensuring functionality of the device.
6 7 8		Drivers have been certified to be compatible with specified operating system versions for their respective device classes. The catalog file protects the integrity of the driver.
9		Microsoft distributes the Driver Protection List to prevent known bad deriver from being installed.
11		Processing Rights Managed Items
12		Certain drivers (SAP) have been explicitly certified to protect DRM content.
13		MSDN – DRM Overview
14		A DRM-compliant driver must prevent
15		unauthorized copying while digital content is being played. In addition, the driver must
16		disable all digital outputs that can transmit the content over a standard interface (such
17		as S/PDIF) through which the decrypted content can be captured.
18	said representation including:	
19	element information contained within said first rights management data	The elements of a driver package include: A driver that is typically a dynamic-link
20	structure; and	library with the .sys filename extension.  An INF file containing information that the
21		system Setup components use to install support for the device.
22		A driver catalog file containing the digital signature.
23		One or more optional co-installers which are a Win32® DLL that assists in device
24		installation NT-based operating systems. Other files, such as a device installation
25		application, a device icon, and so forth.
26		XP DDK - INF Version Section
27		The LayoutFile entry specifies one or more
28		additional system-supplied INF files that contain layout information on the source
		media required for installing the software

	1	
1		described in this INF. All system-supplied INF files specify this entry.
2		and specification
3		The CatalogFile entry specifies a catalog (.cat) file to be included on the distribution media of a device/driver.
4	organization information regarding	Within an INF is a hierarchy with the top
5	the organization of said elements within said first rights management	being a list of manufacturers, and sub-lists of models and at the bottom a list of install
6	data structure; and	information by model.
_		For Windows XP and later versions of NT-
. 7		based operating systems, entries in the
. 8		Manufacturer section can be decorated to
		specify operating system versions. The specified versions indicate OS versions
9		with which the specified INF Models
10	•	sections will be used. If no versions are
		specified, Setup uses the specified <i>Models</i> section for all versions of all operating
-11		systems.
12		
	·	INF's SourceDisksNames and SourceDisksFiles sections specify
13		organization information.
14		XP DDK Source Media for INFs
		The methods you should use to specify source media for device files depend on
15		whether your INFs ship separately from the
16		operating system or are included with the
		operating system. INFs for drivers that are delivered
17		separately from the operating system
18		specify where the files are located using SourceDisksNames and SourceDisksFiles
10		sections.
19	**	If the files to support the device are
20	•	included with the operating system, the INF must specify a LayoutFile entry in the
_,		Version section of the file. Such an entry
21		specifies where the files reside on the
22		operating system media. An INF that specifies a LayoutFile entry must not
22		include SourceDisksNames and
23	·	SourceDisksFiles sections.
24		XP DDK - INF SourceDisksNames Section
25		A SourceDisksNames section identifies
25		the distribution disks or CD-ROM discs
26		that contain the source files to be
		transferred to the target machine during installation. Relevant values of an entry in
27		the INF include:
28		diskid Specifies a source disk.
H		disk-description - Describes the contents

1		and/or purpose of the disk identified by diskid.
2		tag-or-cab-file This optional value
3		specifies the name of a tag file or cabinet file
		supplied on the distribution disk, either in the installation root or in the subdirectory
4		specified by path, if any.
5		path—This optional value specifies the
_		path to the directory on the distribution disk containing source files. The path is
6		relative to the installation root and is
· 7		expressed as \dirname1\dirname2 and so forth.
8		flags For Windows XP and later, setting
		this to 0x10 forces Setup to use cab-or-tag-
9		file as a cabinet file name, and to use tag- file as a tag file name. Otherwise, flags is
10		for internal use only.
-		tag-file For Windows XP and later, if flags is set to 0x10, this optional value
11		specifies the name of a tag file supplied on
12		the distribution medium, either in the
12		installation root or in the subdirectory specified by path. The value should specify
13		the file name and extension without path
14		information.  XP DDK INF SourceDisksFiles Section
15		A SourceDisksFiles section names the
		source files used during installation, identifies the source disks (or CD-ROM
16		discs) that contain those files, and provides
17		the path to the subdirectories, if any, on the distribution disks containing individual
18		files. Relevant values in an entry in the
10		INF would include:
19		filename Specifies the name of the file on the source disk.
20		diskid Specifies the integer identifying
		the source disk that contains the file. This value and the initial path to the
21		subdir(ectory), if any, containing the
22		named file must be defined in a SourceDisksNames section of the same
23		INF.
. 23		subdir This optional value specifies the
24		subdirectory (relative to the SourceDisksNames path specification, if
25		any) on the source disk where the named
٠	information relating to metadata, said	file resides.
26	metadata including:	
27	metadata rules used at least in part to	The driver manufacture can specify rules in
20	govern at least one aspect of use and/or display of content stored within a rights	the INF that govern the installation and/or use of the driver. For example, security
28	management data structure,	entries specify an access control list for the

driver. Driver developers can specify rules in an INF file that determines behavior of the driver package based on the user's operating system version, including product type and suite. Also, rules related to logging can be specified as mentioned in next claim element.

### For Example - Access Control List Rules

XP DDK - Tightening File-Open Security in a Device INF File For Microsoft Windows 2000 and later, Microsoft tightened file-open security in the class installer INFs for certain device classes, including CDROM, DiskDrive, FDC, FloppyDisk, HDC, and SCSIAdapter.

If you are unsure whether the class installer for your device has tightened security on file opens, you should tighten security by using the device's INF file to assign a value to the DeviceCharacteristics value name in the registry. Do this within an addregistry-section, which is specified using the INF AddReg directive. XP-DDK -- INF AddReg Directive

An INF can also contain one or more. optional add-registry-section.security sections, each specifying a security descriptor that will be applied to all registry values described within a named addregistry-section.

A Security entry specifies a security descriptor for the device. The securitydescriptor-string is a string with tokens to indicate the DACL (D:) security component. A class-installer INF can specify a security descriptor for a device class. A device INF can specify a security descriptor for an individual device. overriding the security for the class. If the class and/or device INF specifies a security-descriptor-string, the PnP Manager propagates the descriptor to all the device objects for a device, including the FDO, filter DOs, and the PDO.

For Example - Operating System Versioning

Operating-System Versioning for Drivers

٠.	.	
]		under Windows XP
3		Setup selects the [Models] section to use based on the following rules:
· 4		If the INF contains [Models] sections for several major or minor operating system version numbers, Setup uses the section
6		with the highest version numbers that are not higher than the operating system version on which the installation is taking place.
7		piace.
. 9		If the INF [Models] sections that match the operating system version also include product type decorations, product suite decorations, or both, then Setup selects the
10		section that most closely matches the running operating system.
. 11	said metadata rules including at least one rule specifying that information	The AddService directive can set up event-logging services for drivers.
12	relating to at least one use or display of said content be recorded and/or	INF AddService Directive An AddService directive is used to control
13	reported.	how (and when) the services of particular Windows 2000 or later device's drivers are
14		loaded, any dependencies on other
15		underlying legacy drivers or services, and so forth. Optionally, this directive sets up
16		event-logging services by the devices/drivers as well.  Relevant sections of the directive's entry
17		include: event-log-install-section -Optionally
18		references an INF-writer-defined section in which event-logging services for this
19		device (or devices) are set up.  EventLogType Optionally specifies one
20		of System, Security, or Application. If omitted, this defaults to System, which is almost always the appropriate value for the
21		almost always the appropriate value for the installation of device drivers. For example,
22		an INF would specify Security only if the to-be-installed driver provides its own
23		security support.  EventName Optionally specifies a name
24		to use for the event log. If omitted, this defaults to the given ServiceName.
25		The state of the s
26	26 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	
27	35. A descriptive data structure as in claim 34, in which:	
28	said first rights management data structure comprises a first secure container.	The driver package is secured through a catalog file that is signed by Microsoft's Windows Hardware Quality Lab and
		· · · · · · · · · · · · · · · · · · ·

.1		contains the hash of each file of the driver's package. The INF identifies the catalog file used to sign the driver package.
3		1 me ases to sign the arrow paeriage.
. 4	36. A descriptive data structure as in claim 35, in which:	
5	said first secure container comprises:	The first secure container is the driver package secured by a catalog file.
6	said content; and	The content is the driver and related files within the signed driver package.
7	rules at least in part governing at least one use of said content.	The rules are within the INF, which is part of the signed driver package.
8 · 9	37. A descriptive data structure as in claim 36, wherein the descriptive data structure is stored in said first secure container.	The INF is stored within the signed driver package.
,	Stored in Said That Scoule communer.	
10	44. A descriptive data structure as in claim 34, further including:	
11	a representation of the format of data contained in a second rights management	The manufacture and models sections in the INF Version section are provided for
12	data structure,	the possibility of a single INF representing the format for multiple drivers.
13		Operating system version "decorating"
14		relating the architecture, major and minor operating systems versions, product and
15 16		suit information all relate to the target environment and is used to identify the files necessary for the target environment.
17		An INF file, such as in the case of
18		operating system targeting, can be used for more than one driver package since it can contain more than one catalog file.
19		Further an INF can address the drives
20	said second rights management data	necessary for a multi-functional device.  The files of the second data structure would
21	structure differing in at least one respect from said first rights management data	vary from the files on the first data structure.
22	structure.	
23	45. A descriptive data structure as in claim 44, in which:	
24	said information regarding elements contained within said first rights	INF specify where the driver files are located using the SourceDiskNames and
25	management data structure includes	SourceDiskFiles sections.
26	information relating to the location of at least one such element.	
27	46. A descriptive data structure as in claim	
28	44, further including: a first target data block including information relating to a first target	Operating system version "decorating" relating the architecture, major and minor
	E	vhihis Pi

environment in which the descriptive data structure may be used.	operating systems versions, product and suit information all relate to the first target environment.
47. A descriptive data structure as in claim 46, further including:	
a second target data block including information relating to a second target environment in which the descriptive data structure may be used.	Operating system version decorating will cover multiple operating systems.
said second target environment differing in at least one respect from said first target environment.	This is the reason for version decorating.
48. A descriptive data structure as in claim 46, further including:	
a source message field containing information at least in part identifying the source for the descriptive data structure.	The provider entry in the version section of the INF identifies the provider of the INF file. Also, the INF contains a manufacture
	section.
	·

	<u> </u>		
4	製能	GLAIMICANGUAGE AND THE	THE CLAIM OF THE RINGEMENT AND THE PROPERTY OF
· 5	58		Product Infringing: Microsoft Reader SDK and Microsoft Digital Asset Server.
		method of creating a first secure	Method is carried out by Microsoft's
6		ntainer, said method including the lowing steps:	Digital Asset Server and Microsoft's Litgen tools
7	(a)	accessing a descriptive data structure,	opf file describing the file structure of a protected e-book including metadata,
8		said descriptive data structure including or addressing	manifest, and "spine" information
9		(1) organization information at least in part describing a required or	Organization information regarding organization of the ebook and the
0	·	desired organization of a content section of said first secure container, and	inscription as specified in the manifest and spine information in the .opf file
1		(2) metadata information at least in part specifying at least one step	Metadata constitutes rules specifying the degree of security to use and/or XrML
2		required or desired in creation of said first secure container;	rules
3	(b)	organize said first secure container	e-book packaging carried out by Microsoft Litgen tool
4	(c)	contents using said metadata information to at	Step performed by Digital Asset Server;
5	(9)	least in part determine specific	example of specific information is
		information required to be included in	owner/purchaser information required in
6		said first secure container contents; and	the inscription process
7	(d)	generating or identifying at least one rule designed to control at least one	Analyzing the metadata and finally packaging the e-book using a particular
8		aspect of access to or use of at least a portion of said first secure container	security level specified through the metadata
9		contents.	
<b>.</b> ∄		A method as in claim 58, in which:	
0	(a)	said specific information required to be included includes information at	Owner purchaser information required in the inscription process; XrML rule
1 -		least in part identifying at least one	requiring display of copyright notice
2		owner or creator of at least a portion of said first secure container contents.	
Ħ			

•••		
5	58.	Product Infringing: All products that host the Microsoft Common Language Runtime or Compact Common Language Runtime.
6	A method of creating a first secure container, said method including the	Method is practiced by a user using the Common Language Runtime (CLR) or
7		Compact Common Language Runtime
	following steps;	(CCLR) to create a dynamic shared
8		assembly or .NET Framework SDK to
٨		create a shared assembly
9	(a) accessing a descriptive data structure,	.NET framework Assembly class and/or
10	said descriptive data structure	AssemblyBuilder class and/or
10	including or addressing	AssemblyInfo file
11	(1) organization information at least	This information is specified in the classes
	in part describing a required or	named above and in the AssemblyInfo file.
12	desired organization of a content	
	section of said first secure	
13	container, and	This information is addressed in the classes
., 1	(2) metadata information at least in part specifying at least one step	and the AssemblyInfo file, e.g., for a shared
14	required or desired in creation of	assembly metadata will be specified that
15	said first secure container;	the assembly is to be signed using specified
15	Said 1115t Secure containers	key
16	(b) using said descriptive data structure to	This step is carried out by applications and
10	organize said first secure container	tools using the classes and assembly info
17	contents;	file, including CLR (or CCLR) and .NET
		Framework SDK
18	(c) using said metadata information to at	This step is carried out by applications and
	least in part determine specific	tools using the assembly info file and classes that specify the metadata required
19	information required to be included in	in the target assembly
	said first secure container contents;	in the target assembly
20	and (d) generating or identifying at least one	User may specify rules, as specified in the
21	rule designed to control at least one	NET Framework SDK, to be placed in the
21	aspect of access to or use of at least a	assembly manifest including such rules
22	portion of said first secure container	requiring that all code be managed (CLR or
	contents.	CCLR compliant), "Code Access Security"
23		permissions be supplied for use of code
		supplied in the assembly, etc
24	64. A method as in claim 58, in which:	
Ī	(a) said creation of said first secure	Can be a server, PC or workstation running
25	container occurs at a first data	CLR (or CCLR) to create a dynamic shared
	processing arrangement located at a	assembly or .NET Framework SDK to
26	first site;	create a shared assembly)
	(b) said first data processing arrangement	Included in virtually any computer
27	including a communications port; and	
	(c) said method further includes:	Developed of the passentheinfo file and/or a
28	(1) prior to said step of accessing said	Download of the assemblyinfo file and/or a
	descriptive data structure, said	file containing a class calling the
١ ا	· ·	

Exhibit B

1

2

1	first data processing arrangement	DefineDynamicAssembly methods or
2	receiving said descriptive data	download of SDK containing
- 2	structure from a second data	assemblybuilder class from a second site
3 .	processing arrangement located at a second site,	/
4	(d) said receipt occurring through said first data processing arrangement	Communications port is normally used for downloading
5	communications port.	
	67. A method as in claim 64, further comprising:	
6	at said first processing site, receiving said	Download of the AssemblyInfo file and/or
7	metadata through said communications	a file containing a class calling the
<i>.</i> .	port.	DefineDynamicAssembly methods or
8.		download of SDK containing assemblybuilder class from a second site
. 9	68. A method as in claim 67, in which,	
	(a) said metadata is received separately	Method practiced when metadata names are
10	from said descriptive data structure.	addressed by the assembly class and a template for the AssemblyInfo file, and
11		values corresponding to those names are
11.		received through a user interface such as
12		provided by Microsoft Visual Studio or are
10	71. A method as in claim 58, in which:	provided from a separate file
13	(a) said specific information required to	The Assembly class definition includes
14	be included includes information at	attributes for company name and trademark
	least in part identifying at least one owner or creator of at least a portion of	information, and these may be required attributes specified in the AssemblyInfo file
15	said first secure container contents.	attributes specified in the rissembly me me
16	72. A method as in claim 58, in which:	
	(a) said specific information required to	The Assembly class definition includes an
17	be included includes a copyright notice.	attribute for copyright field that may be required by the AssemblyInfo file
18	nonec.	
19		
20		
.		
21		•
22		
23		
24		
24		
25		
1	•	the state of the s
		•
26		

E GLAIMIFANGUAGE OF S	CLAIMODINERINGEMENTE
58.	Product Infringing: Microsoft .NET Framework, Visual Studio .NET, and tools that include the Assembly Generator tool AL.exe.
A method of creating a first secure container, said method including the following steps;  (a) accessing a descriptive data structure, said descriptive data structure including or addressing	The Assembly Generation tool generates a portable execution file with an assembly manifest from one or more files that are either Microsoft intermediate language (MSIL) modules or resource files. When using the tool's signing option, the assembly becomes a secure container.  The descriptive data structure is the text file used as input by the Assembly Generation tool.
(1) organization information at least in part describing a required or desired organization of a content section of said first secure container, and	The DDS specifies the link and or embed directives to indicate which source files should be included in the assembly, how the included resource will be tagged, and if the resource will be private. Private resources are not visible to other assemblies.  These tags are used to organize the assembly into named sections.  Private attributes are used to organize the assembly into both public and private sections. (Public sections are the default.)
(2) metadata information at least in part specifying at least one step required or desired in creation of said first secure container;	The text file can contain "options" relating to how the assembly should be built and additional information that should be included.
	Main – Specifies the method to use as an entry point when converting a module to an executable file.  Algid – Specifies an algorithm to hash all files.  Comp – Specifies string for the Company field.  Conf – Specifies string for
	Configuration field  Copy - Specifies string for Copyright field.  Culture - Specifies the culture string to associate with the assembly.  Delay - Variation of this option ispecifies whether the assembly will be

Exhibit B 

.2

.14

	#	
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16		fully or partially signed and whether the public key is placed in the assembly.  Description - Specifies the description field.  Evidence - Embeds file in the assembly with the resource name  Security Evidence.  Fileversion - Specifies the file version of the assembly.  Flags - Specifies flags for such things as the assembly is side-by-side compatible, assembly cannot execute with other versions if either they are executing in the same application domain, process or computer.  Keyf - Specifies a file that contains a key or key pair to sign an assembly.  Keyn - Specifies the container that holds a key pair.  Product - Specifies string for Product field.  Product - Specifies string for Product Version.  Template - Specifies the assembly fro which to inherit all assembly metadata.  Title - Specifies string for Title field.  Trade - Specifies string for Trademark field.  V - Specifies version information.  The following directives are used to specify which files are to be compiled into the assembly, how they will be tagged, and
18 19 20 21 22 23 24 25 26 27 28	(c) using said metadata information to at least in part determine specific information required to be included in said first secure container contents; and	whether or not they will be visible to other assemblies, AKA private:  Embed[name, private] - copies the content of the file into the assembly and applies an optional name tag, and optional private attribute.  Link[name, private] - file becomes part of the assembly via a link and applies an optional name tag, and optional private attribute.  The following are some of the "options" address what information should be included in the secure container:  Main - Specifies the method to use as an entry point when converting a module to an executable file.  Comp - Specifies string for the Company field.  Conf - Specifies string for Copyright

		field.  Culture – Specifies the culture string to associate with the assembly.  Description – Specifies the description field.  Evidence – Embeds file in the assembly with the resource name Security. Evidence.  Fileversion – Specifies the file version of the assembly.
		Flags - Specifies flags for such things as the assembly is side-by-side compatible, assembly cannot execute with other versions if either they are executing in the same application domain, process or computer.  Keyf - Specifies a file that contains a
		key or key pair to sign an assembly.  Keyn - Specifies the container that hold a key pair.  Product - Specifies string for Product field.  Productv - Specifies string for Product Version.  Template - Specifies the assembly fro
(1)		which to inherit all assembly metadata.  Title – Specifies string for Title field.  Trade – Specifics string for Trademark field.  V – Specifies version information.  User may specify rules, as specified in the
(d)	generating or identifying at least one rule designed to control at least one aspect of access to or use of at least a portion of said first secure container contents.	.NET Framework SDK, to be placed in the assembly manifest including such rules requiring that all code be managed (CLR compliant), "Code Access Security" permissions be supplied for use of code supplied in the assembly, etc.
71.	A method as in claim 58, in which:	
(a)	said specific information required to be included includes information at least in part identifying at least one owner or creator of at least a portion of said first secure container contents.	The following "options" specifies owner and creator information:  Comp - Specifies string for the Company field. Copy - Specifies string for Copyright field. Trade - Specifics string for Trademark field.
72.	A method as in claim 58, in which:	
(a)	said specific information required to be included includes a copyright notice.	The copy "option" specifies the string for the for the Copyright field.

	10,000,000	
. 3	ANGUA GENERAL ANGUA ANGUA GENERAL ANGUA	#GLATMIOEINFRINGEMENT
. 4	1.	Products infringing: All products that include the Common Language Runtime or Compact Common Language Runtime or Common
5		Language Infrastructure.
6	A method for using at least one resource processed in a secure operating environment at	Resource may constitute a Microsoft Windows process or hardware element; secure operating
7	a first appliance, said method comprising:	environment is Microsoft Common Language Runtime ("CLR") environment, Common
.8		Language Infrastructure ("CLI") or Compact CLR ("CCLR"); first appliance is computer
9	•	running CLR, CLI or Compact CLR. Two infringing scenarios are set forth herein: (1)
10.		For CLR, an administrator, using the NET framework caspol exe tool remotely configures
11		security policy in a .NET configuration file for a machine, enterprise, user, or application and
12		that security policy interacts with rules or evidence declared in a shared assembly
13		provided by another entity ("1st scenario"); and (2) for CLR, CLI and CCLR two assemblies
14		are delivered to an appliance; the first assembly has a rule that demands permissions
15		from a caller in the second assembly, and the second assembly includes a control that asserts
16		such permissions or provides evidence that convinces the runtime that it has such
17		permissions. ("2 <sup>nd</sup> scenario"). In each scenario Microsoft .NET "Code Access Security"
18		framework or "Role Based Security" framework is used.
.19	(a) securely receiving a first entity's control at	1 <sup>st</sup> scenario: first entity is the administrator,
20	said first appliance, said first entity being	and the policy that constitutes this entity's control is securely received at the first
21	located remotely from said operating environment and said first appliance;	appliance through a session established between the administrator's computer and the
22		first appliance, requiring security credentials such as the administrator's login and password
23		or other secure session means.  2 <sup>nd</sup> scenario: first entity is creator or distributor
24	<u> </u>	of the first assembly, assembly manifest includes a control demanding or refusing or
25		otherwise asserting a security action on permissions from a caller; first assembly is
26		integrity-checked.
27	(b) securely receiving a second entity's control at said first appliance, said second entity being	Second entity's control is contained in shared assembly manifest (and therefore integrity
28	located remotely from said operating environment and said first appliance, said	protected) that provides evidence for obtaining permissions, or asserts permissions; assembly
. 20	second entity being different from said first	creator/distributor is located remotely and is
į		

٠		
1 2	entity; and	not the administrator (1 <sup>st</sup> scenario) or creator/distributor of the first container (2 <sup>nd</sup> scenario);
3	(c) securely processing a data item at said first appliance, using at least one resource,	Secure processing is carried out by CLR, CLI or CCLR, Data item constitutes an executable
4	including securely applying, at said first appliance through use of said at least one	code element, an interface controlled by such an executable, a data collection or stream (such
5 6	resource said first entity's control and said second entity's control to govern use of said data item.	as media file or stream or text file) or an environment variable. CLR, CLI or CCLR securely processes the rules, which will in both scenarios govern access to methods and data
7		from the first assembly. The resource named in the claim is, e.g., a Windows process that is established by the runtime or hardware element
8		on the computer.
9 <sub>.</sub> 10	51. A method as in claim 1 wherein at least said secure processing step is performed at an end user electronic appliance.	Consumer computer or appliance running Microsoft CLR, CLI or CCLR).
11	58. A method as in claim 1 wherein the step of securely receiving a first entity's control	1 <sup>st</sup> scenario 1: link is LAN or WAN; 2 <sup>nd</sup> scenario: link is any telecommunications link,
12	comprises securely receiving said first entity's control from a remote location over a	including the internet.
13	telecommunications link, and the step of securely receiving said second entity's control	
14	comprises securely receiving said second entity's control from the same or different remote location over the same or different	
15	telecommunications link.	
16 17 18	65. A method as in claim 1 wherein the processing step includes processing said first and second controls within the same secure processing environment.	Secure processing environment is CLR, CLI or CCLR running on user's computer or appliance.
19	71. A method as in claim 1 further including	In scenario 2, arrangement consists of the stack frame, and the corresponding array of
20 21	the step of securely combining said first entity's control and said second entity's control to provide a combined control arrangement.	permission grants for assemblies on the stack, and the permission demanded by the first assembly. Secure combining performed by the
22	76. A method as in claim 1 wherein said two	CLR, CLI or CCLR. Steps are performed at different times in both
23	securely receiving steps are independently performed at different times.	scenarios.
24	84. A method as in claim 1 wherein at least one of the first entity's control and the second	In both scenarios the second entity supplies an assembly with a demand procedure executed
25	entity's control comprises at least one executable component and at least one data component.	by the CLR, CLI or CCLR. The data component is a specific attribute value referenced by the assembly.
26	89. A method as in claim 1 wherein said first appliance includes a protected processing	Microsoft Common Language Runtime (CLR), Common Language Infrastructure (CLI), or
27	environment, and wherein:	Compact Common Language Runtime (CCLR) environment.
28	(a) said method further comprises a step of receiving, at said first appliance, said data item	Typically occurs in both scenarios.
·	•	.

	recei	ceiving said first entity's control; and												<u>.                                    </u>				• :	ha (C)	I D	
	separately and at a different time from said receiving said first entity's control; and  (b) said securely processing step is performed at least in part in said protected processing environment									Protec CLI or	ted pi CCL	roces R.	sing	envir	onmen	t is t	ne C	LK			
	envi	on	n pa nent	נונות נ	1 5a	μ	·		u pr	00000				. ,							
$\ $		<del></del>	·	<u>:</u>	·	:		:					L								
$\ $			٠						: ,			•			•	٠.	. :		• .		
									•				•					•		:	
		-												•				:			
												:	• . • .					•			٠,
													· ·		•					•	
				•	•	٠			•					•							
						,				•		•	•			• •					
	-	•		•				-												•	
							•												•		
														•							
				•																	
																	•				
					•					٠											
														•							
I				•						:		•				•					
																				٠	
				•				•	•									•			
							•							٠		•					
											•					•	,				
												• •			•	•					•
						•				· ;					,	•					
																_	•		-		
	•													•	•	•					
																			:		
							٠								•	-		•			
																			•		
					•						•										
1													-								

293482.02

_	FOR U.S. PATE	41 140. 3,702,071
3		•
·4 5	22.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
6 7	A method of securely controlling use by a third party of at least one protected operation with respect to a data item comprising:	A user (third party) accesses an IRM-protected data item governed by IRM controls under two or more RMS servers. For example, the data item may be a IRM-protected document.
9		The IRM controls may be associated with the data item directly or via a IRM-protected container holding the IRM-protected data item,
10		such as an IRM-protected email with the IRM- protected document attached.
11	(a) supplying at least a first control from a first party to said third party;	The user acquires a first use license from a first RMS server (first party) enabling access to, the IRM-protected data item under the IRM rules
12 13		associated with the first RMS server. For example: (1) the first use license from the first
14		RMS server permits the user to access a IRM- protected document contained within or
15		attached to an IRM-protected email; or (2) the first use license from the first RMS server applies a first set of IRM rules to an IRM-
16		protected document. The user acquires a second use license from a
17	(b) supplying, to said third party, at least a second control from a second party different from said first party;	second RMS server (second party) enabling access to the IRM-protected data item under
18	•	the IRM rules associated with the second RMS server. For example: (1) in addition to the user being given access to an IRM-protected
19 20		email based on a first use license, a second RMS server provides a second use license
21		enabling access to the IRM-protected document attached thereto; or (2) the second use license from the second RMS server
22		applies a second set of IRM rules to the IRM- protected document.
23	(c) securely combining at said third party's location, said first and second controls to form	The first and second use licenses are combined to form a control arrangement that governs
24	a control arrangement;	access to the IRM-protected data item.
25 26	(d) securely requiring use of said control arrangement in order to perform at least one protected operation using said data item; and	The combined first and second use licenses govern access to the IRM-protected data item.
27	(e) securely performing said at least one protected operation on behalf of said third	The user performs a protected operation (e.g., read, print, edit) on the IRM-protected data
28	party with respect to said data item by at least in part employing said control arrangement	item. The combined first and second use licenses are employed to permit the protected operation.

.		and protected by
1	23. A method as in claim 22 wherein said data item is protected.	The data item is encrypted and protected by IRM.
2	39 A method as in claim 22 further including	The first and/or second use license are securely
3	securely and persistently associating at least	and persistently associated with the IRM- protected data item.
	one of: (a) said first control, (b) said second control, and (c) said control arrangement, with	protection during the protection of the protecti
4	said data item.	Steps performed at a user's computer or
5	53. A method as in claim 22 wherein at least two of the recited steps are performed at an end	appliance.
6	user electronic appliance.  60. A method as in claim 22 wherein step (a)	The first and second use licenses are received
7	comprises supplying said first control from at	over a telecommunications link such as a networking or modem/serial interface.
	least one remote location over a telecommunications link, and step (b)	
8	comprises supplying said second control from	
9	the same or different remote location over the same or different telecommunications link	
10	67 A method as in claim 22 wherein at least	Steps are performed at user's computer or appliance.
	step (c) is performed within the same secure processing environment at said third party's	upp
11	location.	
12	91. A method as in claim 22 wherein: (a) said method further comprises supplying	The first use license (first control) is received
13	said data item to said third party separately and	at the time that the user accesses the data item, which occurs separately and at a different time
	at a different time from supplying of said first control to said third party; and	from receipt of the IRM-protected data item
14	Collidor to said and be- 3,	itself.
		The protected operations require decryption of
15	(b) said securely performing step comprises	The protected operations require decryption of the protected content, which is done inside the
15 16	(b) said securely performing step comprises performing said protected operation at least in part in a protected processing environment.	The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by
16	linerforming said protected operation at least in	The protected operations require decryption of the protected content, which is done inside the
	linerforming said protected operation at least in	The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
16	linerforming said protected operation at least in	The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
16 17	linerforming said protected operation at least in	The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
16 17 18 19	linerforming said protected operation at least in	The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
16 17 18 19 20	linerforming said protected operation at least in	The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
16 17 18 19	linerforming said protected operation at least in	The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
16 17 18 19 20	linerforming said protected operation at least in	The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
16 17 18 19 20 21 22	linerforming said protected operation at least in	The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
16 17 18 19 20 21 22 23	linerforming said protected operation at least in	The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
16 17 18 19 20 21 22	linerforming said protected operation at least in	The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
16 17 18 19 20 21 22 23 24	linerforming said protected operation at least in	The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
16 17 18 19 20 21 22 23 24 25	linerforming said protected operation at least in	The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
16 17 18 19 20 21 22 23 24	linerforming said protected operation at least in	The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
16 17 18 19 20 21 22 23 24 25	linerforming said protected operation at least in	The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
16 17 18 19 20 21 22 23 24 25 26	linerforming said protected operation at least in	The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-

	101010111	121.11 110.0,502,051
3	26.	Products infringing: Visual Studio.NET,
	20.	.NET Framework SDK, and all products
4		that include the Common Language
	•	Runtime or Compact Common Language
5		Runtime or Common Language
		Infrastructure.
6	A secure method for combining data	
7	items into a composite data item	
' ]	comprising:	A first signed and licensed .NET
8	(a) securely providing, from a first location	component, NET assembly, managed
	to a second location, a first data item	control and/or Web control (component) is
9	having at least a first control associated	the first data item. The first .NET
ł	therewith;	component developer (first location)
10		provides the application assembly
		developer (second location) the first
11		component. The first control is the set of
12		declarative statements comprising the
12		LicenseProviderAttribute (alternately
13		referred to as license controls).
13	(b) securely providing, from a third	A second signed and licensed component is
14	location to said second location, a second	the second data item. The second component developer (third location)
	data item having at least a second control	provides the application assembly
15	associated therewith;	developer (second location) the second
		component. The second control is the set
16	·	of declarative statements comprising the
,,,		LicenseProviderAttribute.
17	(c) forming, at said second location, a	The application assembly developer will
18	composite of said first and second data	include at least the two components into its
10	items:	assembly.
19	(d) securely combining. at said second	At the second location, the application
	location, said first and second controls to	assembly developer uses the .NET runtime
20	form a control arrangement; and	that includes the LicenseManager.
		Whenever a component is instantiated
21		(here, an instance of the first licensed
		component), the license manager accesses
22	·	the proper validation mechanism for the
<b>.</b>		component. The license controls (first
23		control) for the runtime license (derived
24		from the design time license) are bound
24		into the header of the .NET application
25		assembly, along with the second control for
23		the second component.
.26		10 10 NET accomply handles the
		Visual Studio.NET securely handles the creation of runtime license controls.
27		Runtime licenses are embedded into (and
		bound to) the executing application
28		assembly. The license control attribute
••		assembly. The needse condet addresse

. 2		included in the first component is customized in the second location to
رت 3.		express and require the runtime license. In a more advanced scenario, the License
4		Complier tool can be used to create a ".licenses file" containing licenses for
5		multiple components, including-runtime licenses for components and classes created
		by the license provider. This .licenses file is embedded into the assembly.
6 7		The third control set comprises the runtime
		license controls for the first and second components (that had been bound to the
8		assembly), the declarative controls provided by the application assembly
9		developer, and any runtime licenses for other components included by the
10		developer in application assembly. The controls are typically integrated into the
11		header of the .NET application assembly calling the first licensed component.
12	(e) performing at least one operation on said composite of said first and second data	The proper execution of the application will require that the assembly have run
13	items based at least in part on said control arrangement.	time licenses for the two components.
14	111111111111111111111111111111111111111	
15	27. A method as in claim 26 wherein said	The set of declarative statements comprising the LicenseProviderAttribute of
16	combining step includes preserving each of said first and second controls in said composite set.	both the first and second components are included in the application assembly.
17		:
	28. A method as in claim 26 wherein said performing step comprises governing the	The application will require the first and second controls to operate properly when it
18	operation on said composite of said first	calls the first and second data items,
19	and second data items in accordance with said first control and said second control.	respectively.
20	20 4 4 3 - 1 - 1 - 26 - 1 - 1 1	Ciaring the commonant that has amhadded
21	29. A method as in claim 26 wherein said providing step includes ensuring the integrity of said association between said	Signing the component that has embedded within it the license control ensures the integrity of the association of the control
22	first controls and said first data item is maintained during at least one of	and data item.
23	transmission, storage and processing of said first data item.	
24		
25	31. A method as in claim 26 wherein said providing step comprises codelivering said	The component includes the license control and therefore they are codelivered.
26	first data item and said first control.	
20.	40. A method as in claim 26 further	Each component includes the license
27	including the step of securely ensuring that	control. Signing the component that has embedded within it the license control
28	at least one of (a) said first control, (b) said second control, and (c) said control	ensures the persistence of the association of
İ	arrangement, is persistently associated with	the control and data item.
- 1		#

1	at least one of said first and second data items.	
2		At least step (e) is typically performed at an
3	54. A method as in claim 26 wherein at least one of steps (c), (d) and (e) is	end-user electronic appliance.
.4	performed at an end user electronic appliance.	
5		Microsoft maintains Web sites where a
6	61. A method as in claim 26 wherein step (a) comprises providing said first data item from at least one remote location over a	developer can get components over the Web. These sites include references
7	telecommunications link, and step (b)	whereby a developer may obtain components through their Web connection.
8	comprises providing said second data item from the same or different remote location	One such site is Internet Explorer web  Control Gallery at
9	over the same or different telecommunications link.	ie.components.microsoft.com/webcontrols
10	68. A method as in claim 26 wherein step (d) is performed within the same secure	Typically, step (d) will be performed within the same secure processing
11	processing environment at said second location.	environment.
12	79. A method as in claim 26 wherein steps	The application assembly developer will
13	(a) and (b) are performed at different times.	typically acquire components at different times.
14 15 16	86. A method as in claim 26 wherein at least one of the first and second controls comprises at least one executable component and at least one data	The component must include an executable and can include a data items as a EULA, readme file or help file.
15	least one of the first and second controls	and can include a data items as a EULA,
15 16	least one of the first and second controls comprises at least one executable component and at least one data	and can include a data items as a EULA,
15 16 17	least one of the first and second controls comprises at least one executable component and at least one data	and can include a data items as a EULA,
15 16 17 18	least one of the first and second controls comprises at least one executable component and at least one data	and can include a data items as a EULA,
15 16 17 18 19 20	least one of the first and second controls comprises at least one executable component and at least one data	and can include a data items as a EULA,
15 16 17 18 19 20 21	least one of the first and second controls comprises at least one executable component and at least one data	and can include a data items as a EULA,
15 16 17 18 19 20 21 22	least one of the first and second controls comprises at least one executable component and at least one data	and can include a data items as a EULA,
15 16 17 18 19 20 21 22 23	least one of the first and second controls comprises at least one executable component and at least one data	and can include a data items as a EULA,
15 16 17 18 19 20 21 22 23 24	least one of the first and second controls comprises at least one executable component and at least one data	and can include a data items as a EULA,
15 16 17 18 19 20 21 22 23	least one of the first and second controls comprises at least one executable component and at least one data	and can include a data items as a EULA,
15 16 17 18 19 20 21 22 23 24	least one of the first and second controls comprises at least one executable component and at least one data	and can include a data items as a EULA,

3		· ·
	CHAIMTEANGUAGE SUSSE	整数整GEATM!OF INFRINGEMENTS: Fixe
4	35	Infringing products include: Windows Media Player, Individualized DRM Clients
5		and the Secure Audio Path (SAP) technology.
.6	A method for using at least one resource	teciniology.
7	processed by a secure operating environment, said method comprising:	
.8	securely receiving a first load module provided by a first entity external to said	The Individualized DRM Client (first load module) is a signed security upgrade DLL.
9	operating environment	It is also bound to the hardware ID of the machine on which it runs. It is therefore
10	securely receiving a second load module	securely delivered and integrity protected.  A SAP certified driver is also signed and
11	provided by a second entity external to said operating environment, said second entity	carries with it a certificate that indicates its compliance with SAP criteria. If it is
12	being different from said first entity; and	delivered to a PC it is secure in the sense that it is integrity protected. This driver
13 -		would not come from the same entity as the Individualization DLL.
14	securely processing, using at least one resource, a data item associated with said	If a WM audio file targeted to the Individualized DRM client carries with it a
15	first and second load modules, including securely applying said first and second load	requirement that SAP be supported to render the WMF contents, the content is
16	modules to manage use of said data item.	processed for playing through a soundcard using the WMP and by applying the DRM
17		client - which decrypts the content and negotiates with the DRM kernel processing
18		of the content through a Secure Audio Path that includes the SAP-certified audio
19		driver.
	50 A di la cialità 25 mb acciont	All steps occur at the user's PC that
20	56. A method as in claim 35 wherein at least two of the recited steps are performed	supports the WMP and DRM client and SAP.
21	at an end user electronic appliance.	DAI.
22	63. A method as in claim 35 wherein said first load module receiving step comprises	The Driver and DRM client are received from distinct locations and may be
23	securely receiving said first load module from at least one remote location over at	delivered securely over the Internet. They are delivered securely in that each is
24	least one telecommunications link, and said second load module receiving step	integrity protected.
25	comprises securely receiving said second	
26	load module from the same or different remote location over the same or different	
27	telecommunications link.	D 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
28	70. A method as in claim 35 wherein said securely processing step comprises securely executing said first and second	Both load modules are executed on the PC within the WMP/DRM Client/SAP environment.
· • •		

----Exhibit B

1	CANAL CEATMIE ANGUAGE SERVE	MESSAGEATMIOFINERINGEMENTERS
2	load modules within the same secure processing environment.	
3		Since both the DRM client and the driver
4	74. A method as in claim 35 further including securely combining said first and	are DLLs in the same audio rendering
.4.	second load modules to provide a	chain, they exist as an execution
5	combined executable.	environment.
.6	81. A method as in claim 35 wherein said	The driver and Individualization DLL need
7	securely receiving steps are performed	not be received at the same time.
-	independently at different times.	
.8	94. A method as in claim 35 wherein said	The Windows Media Player together with the Individualized DRM Client and Secure
9	secure operating environment includes a protected processing environment, and	Audio Path comprise a protected
10	wherein:	environment for processing protected media. The protected Windows Media
	said method further comprises receiving a	Files are received after the load modules
11	data item within said secure operating environment;	have been received and installed (licenses cannot be acquired until load modules are
12		in place). The processing of the Windows Media File occurs in the protected
13	said first load module receiving step is performed separately and at a time different	environment.
14	from receiving said data item; and	
· '	said securely processing step is performed	
15	at least in part in said protected processing environment.	
16		
17	Examples of SAP-certified drivers include -	as indicated at
18	http://www.microsoft.com/Windows/windov	vsmedia/WM//DRM/FAQ.asp#Security/
19	All VIA controllers with AC-97 codecs	
	All ALI controllers with AC-97 codec	
20	<ul> <li>Intel ICH controllers with AC-97 codecs</li> </ul>	
21	Creative Labs SoundBlaster16/AWE32/AWI	E64/Mihra
22		2047 V 101a
23	Yamaha OPL3	•
	Yamaha DS-1	
24	<ul> <li>Cirrus Logic (Crystal) CS4280</li> </ul>	
25	<ul> <li>Cirrus Logic (Crystal) CS4614 / CS4624</li> </ul>	
26	ESS Maestro 2E	
27	USB Audio	

Cirrus Logic (Crystal) CS4281

1	
2	
· 3	
4	
• 5	
6	
7	
8	
9	
10	
11	
12	
13	
14.	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	

	1	
•	2	
	3	
	4	
	5	
	6	
	7	
	8	
	9	
1	0	
1	1	
	2	
1	3	
1	4.	
1	5	
1	6	
1	7	-
1	8	
1	9	
2	0	
2	1	
2	2	
2	3	
2	1°234.5	
2	:5	
2	:6	
2	27	

,	All SiS controllers with AC-97 codecs
,	Ensoniq ES1370
•	NeoMagic NM6
1	Ensoniq ES1371/73 and CT5880
,	SoundBlaster Live!
	Aureal 8810
ı	Aureal 8820
,	Aureal 8830
,	Conexant Riptide
	ESS Maestro
ı	ESS ISA parts
ı	NeoMagic NM5
	•
	· · · ·

3	FOR U.S. PATENT NO. 5,982,891	
4		
7	36.	Product Infringing: Any product using Common Language Runtime (CLR), Common
5		Language Infrastructure (CLI), or Compact
6		Common Language Runtime (CCLR)  Microsoft CLR, CLI or CCLR (operating
7	A secure operating environment system for managing at least one resource comprising:	environment system), managing any of the resources on a typical computer, including
		memory, files system, communications ports,
8		storage devices, and higher level resources that may use any of these or combinations of them.
9	(a) a communications arrangement	Communications port and Microsoft Internet
10		Protocol stack that may optionally use Secure Socket Layer protocol or IPSEC packet
11		security protocol, supplied with Microsoft Windows.
12	(1) that securely receives a first control	Rule or evidence contained in the manifest of a
	of a first entity external to said operating environment, and	shared assembly, distributed by a first entity that can be used by the CLR, CLI or CCLR to
13	opolating the monanting that	determine permissions that may be needed to cause operations on a data item or resource
14		controlled by another entity; shared assembly
15		is tamper-protected and may be received using secure SSL or IPSEC protocol.
16	(2) securely receives a second control	Rule specified in the manifest of a second
	of a second entity external to said operating environment, said second	shared (Tamper protected) assembly, that demands permissions of callers of its methods.
17	entity being different from said first entity; and	
18	(b) a protected processing environment,	CLR, CLI or CCLR, connected to (e.g.)
19	operatively connected to said communications arrangement, that:	communications port
20	(1) [] securely processes, using at least	CLR, CLI or CCLR uses type safety
	one resource, a data item logically associated with said first and second	mechanisms, access controls, integrity detection, and separation of domains. Data
21	controls, and	item may be any data item that is managed by
22:		the second assembly, which may be a member of such assembly, and whose state or value
23		may be accessible through an interface to other
. '		assemblies, and which is referenced by the first assembly.
24	(2) [] securely applies said first and	CLR, CLI or CCLR processes the demand for
25	second controls to manage said resource for controlling use of said data	permissions from the second assembly, collects the evidence or processes the rule from the first
26	item.	assembly, and determines whether the first
		assembly has the permissions to use the resource to operate on the data item controlled
.27		by the second assembly.
28	57. A system as in claim 36 wherein said protected processing environment is part of an	Computer or electronic appliance running CLR, CLI or CCLR
	processes processing strength	
1	<b>.</b> .	

Exhibit B 43

. 1

İ	end user electronic appliance.	
. 2	64. A system as in claim 36 wherein said communications arrangement receives said	Shared assemblies are designed to be received remotely, e.g., over the internet.
3	first and second controls from at least one remote location over at least one	
4	telecommunications link.  75. A system as in claim 36 wherein said	Arrangement consists of the stack frame and
5	protected processing environment combines	and the corresponding array of permission
6	said first and second controls to provide a combined control arrangement.	grants for assemblies on the stack, and the permission demanded by the second assembly.
7	82. A system as in claim 36 wherein said	Assemblies, including controls, are designed
7	communications arrangement independently	for independent delivery.
8	receives said first and second controls at different times	7.
9	88. A system as in claim 36 wherein at least one of the first control and second controls	The second entity supplies an assembly with a demand procedure (executed by the CLR, CLI
10	comprises at least one executable component and at least one data component.	or CCLR) that includes reference to a specific attribute value (the data component), and the
11	and at least one data component.	protected processing environment executes the
12		executable component (demand) in a manner that is at least in part responsive to the data
		component (execution is in response to the security action supplied in the data item).
13		Security action supplied in the data resily.
14		
15		
16		
17		
	·	•
18		
19		
20		
21		
22		
. 1		
23		
24		
25		
26		• • · · · · · · · · · · · · · · · · · ·
	·	
27		
28		

4	E SE CHAIMEANGUAGEAS E	EMESSICISATMION SINGRINGEMENT AND ASSESSED.
5	36.	Infringing Product: My Services
6	A secure operating environment system for managing at least one resource	Secure operating environment is the secure server for any NET My Services service
7	comprising: a communications arrangement that	(e.g. My Calendar, My Inbox) Secure server receives communications
8	securely receives	formatted using the SOAP-SEC, the security extension to SOAP that is used by
9		My Service servers to receive controls.
10		
11	a first control	The first control is a roleTemplate
12		associated with the service. The roleTemplate identifies specific actions
13		(e.g. read, replace) that can be performed against a certain scope (resource or set of
14		resources).
15	of a first entity external to said operating	The first entity is the administrator of the
16	environment,	server database, or other entity with authority over its content that sets up the
17		roleTemplates and scopes. That entity is independent from and located remotely from the secure server.
18	and securely receives a second control	A role element specified by a specific end user, which is securely received by the
19 20		secure server using the SOAP-SEC protocol.
20		protection
21		
22	of a second entity external to said operating environment, said second entity	The end user is located remotely from the secure server.
23	being different from said first entity;	The protected processing environment is
24	and a protected processing environment, operatively connected to said	the .NET security service (authorization system) operating within the server. The
25	communications arrangement, that:	server uses the SOAP-SEC communication protocol to receive
26		controls.
27	(a) securely processes, using at least one resource, a data item logically associated	"Securely processes" is performing the requested operation on secure server
28	with said first and second controls, and	running .NET. The system will perform the requested operation ensuring that the user
	·	has no access to information outside the
	1	4

Exhibit B 45

-1		·
1		scope computed.
2 3		The resource is the server software and/or hardware used to process the two controls and user data.
4		The first control is the roleTemplate for the service. The second control is the role element for an individual user.
6	d:	The data item is the end user's stored
7		content (e.g. calendar, email inbox, etc.).
8	(b) securely applies said first and second	The secure server determines the result
9	controls to manage said resource for controlling use of said data item.	scope (visible node set) for the operation that is computed from the role element and the roleTemplate. That result scope is used
10		to manage the data item.
11		
12	64. A system as in claim 36 wherein said	The remote location is the site where the user's or administrator's application is
13	communications arrangement receives said first and second controls from at least one	running.
14	remote location over at least one telecommunications link.	The telecommunication link can be the Internet, intranet, VPN or other similar
15		channels.
16	75. A system as in claim 36 wherein said	The role scope incorporating the role element and the role Template.
17	protected processing environment combines said first and second controls to	Clement and die 1939
18	provide a combined control arrangement.	
19	82. A system as in claim 36 wherein said communications arrangement	Administrator and user controls will ordinarily be received at different times.
20	independently receives said first and second controls at different times.	
21	95. A secure operating environment system	This is the normal case for .NET My
22	as in claim 36 wherein said	Services. The user's content is normally stored and updated independently of the
23	a data item separately and at a different time from at least one of said first control	setting of scope elements, role elements and roleTemplates.
24	and said second control.	
25		•
26		•

27

3	·	,
. 4	TAX STATISTICAL TO THE STATE OF	ESCALA CLAIMOFINERINGEMENDE COMPANY
	CHAIMEAING GAG DISC SEAL STATE OF THE SEAL STATE	Product Infringing: Windows CE for Automotive
5	1. A security method comprising:	WCEIA is Microsoft Windows CE for Automotive,
6		sometimes also known by its former name, AutoPC 2.0.
		With WCEfA an OEM can assign their device to a class
7		that only accepts certain kinds of software. The device can be set to accept 1) any software with the correct
. 8		processor/version 2) only certified software or 3) only software from the OEM or Microsoft. These Security (or
9	·	Trust) levels also control to which kernel APIs and
10		middleware APIs the software has access.
10	·	Background:
11		"Microsoft Software Install Manager (SIM), a
12		component of WCEfA, allows you to control what can be installed on your device platform. You can define
- 1		your platform as being open, closed or restricted to new
13		installations, and SIM will enforce these designations." (D,pg.1)
14		
15		"Anything can be installed on an open platform, as long as the applications are compiled for the appropriate
		processor. At the other extreme, no third-party software
16		can be installed on a closed platform. Only certified applications can be installed on a restricted platform."
17	;	(D, pg.1)
18		"By restricting installations to compliant applications,
		the risk of installing and using incompatible or harmful
19		software is greatly reduced, while still keeping the device open for robust, quality applications that enhance
20		the user experience." (F, pg.1)
21		WCEfA also has a Security Layer whose purpose is to
• •		"Create an abstraction layer of security surrounding ISV
22		applications to limit and/or deny access to key Windows CE kernel API calls and WCEfA middleware APIs." I,
23		pg. 1)
24		
	(a) digitally signing a first load module with a	A first load module is a WCEfA software component in a signed .PE file. The first device class is a device that
25	first digital signature designating the first load module for use by a first device class;	only allows software designated as "restricted" (or
26	induite for use of a mist device come,	higher) to be installed. "Restricted" software is software that has been certified. With restricted software, the
		device also implements a Security Layer functionality
27		that limits the kernel and WCEfA API calls that the
28		software can make.
•		
	\$1	!!

Exhibit B 47

1

1.	"SIM Level: I = Restricted
2	Description: Only properly certified CEI (WCEfA device installation) files can be installed on the device.
3	Remote execution is restricted to executables with master key.
4	Key: Logo certified CEI file required. CEI files or EXEs with master keys permitted." (F, pg. 1)
5	"The kernel loader calls it each time a module is loaded
6	by Windows CE. It returns one of the following values that determine the module's access to kernel resources:
7	Value
.8	Meaning
9	OEM_CERTIFY_TRUST (2) The module is trusted by the OEM to perform any operation.
10	OEM_CERTIFY_RUN (1)
11	The module is trusted by the OEM to run but is restricted from making certain function calls.
12	
13	OEM_CERTIFY_FALSE (0) The module is not allowed to run.
14	" (H, pg. 1)
15	Digitally signing: "Before the kernel loads a file, it uses the OEMCertifyModule function to verify that the file
16	contains the proper signature." (N, pg.1)
17	"Signfile.exe: This tool signs an executable with a supplied private key. You can use the following
18	command parameters with this tools AttribString, specifies an optional attribute string to be included in the
19	signature. For example, you could add a string to indicate the trust level of the application." (O. Pg. 1)
20	In the MSDN article Verifying the Signature, the sample
21	code segment states
22	"//the file has a valid signature // we expect the trust level to be returned as signed
23	data //case 'R': dwTrustLevel = OEM_CERTIFY_RUN" (N,
24	pg.2)
25	"The WCEfA Security Layer isolates installed
26	applications from making unrestricted kernel and WCEfA API calls. This allows the OEM to assign one of
	three levels of security to applications and drivers
27	installed in RAM when they are loaded into the system.  The three levels are Trusted, Restricted, and
28	BlockedOn the systems level, the WCEfA Security

i		
1		layer fits between ISV applications and isolates these software modules from having free access to all WinCE
2		kernel calls and WCEfA middleware APIs." (I, pg. 1)
3		The developer submits their application for certification. If it passes, then the .cei file (a form of cab file) receives
4		a certification key from the certifier. The signed PE is
5		within this .cei file.
6	(b) digitally signing a second load module with	A second load module is a WCEfA software component is a signed PE file. The second device class with a
7	a second digital signature different from the first digital signature, the second digital	different tamper resistance or security level is a device that is "Closed", that is, it will not allow third party to
	signature designating the second load module for use by a second device class having at least	software to be installed. A closed device only allows
8	one of tamper resistance and security level different from the at least one of tamper	trusted software to run. The Security Layer setting of "Trusted" allows the Microsoft and OEM software full
9	resistance and security level of the first device	access to kernel and middleware APIs.
10	class;	In the MSDN article <u>Verifying the Signature</u> , the sample code segment states
11		"//the file has a valid signature // we expect the trust level to be returned as signed
. 12		data
13		//case 'T': dwTrustLevel = OEM_CERTIFY_TRUST" (N, pg.2)
14		"Signfile.exe: This tool signs an executable with a
15	·	supplied private key. You can use the following command parameters with this tools AttribString, specifies an optional attribute string to be included in the
16 17		signature. For example, you could add a string to indicate the trust level of the application. (O. Pg. 1)
		"SIM Level: 2 = Closed Description: Platform is limited to software supplied
18		directly by OEM or Microsoft. Third-party applications
19		cannot be installed Key: Master key required for any install or remote
20		execution." (F, pg.1)
21		Related to the Security Layer, the Trusted level "is most likely reserved for MS and OEM applications and
22		drivers." (1, pg. 1)
23		Whereas the .cei files for certified software have a
24	•	certification key (sometimes call MS Logo key), the .cei files from Microsoft or the OEM have a master key
25		attached. ""Master key required for any install or remote execution." (F, p.gl)
26	(c) distributing the first load module for use by	First load module is the certified software from a third
27	at least one device in the first device class; and	party that will be run as part of the "Restricted" first device class.
28		"Once your application is complete, send the .cei file to

1		the organization that is performing validation or
•		certification for the OEM. They would validate it, then
2		either reject or return a cei that has been stamped with a
_		certification key. You would then reproduce this .cei file
· 3		on CD-ROM or a compact flash card and distribute." (D,
,		p.g 5)
4		"APCLoad compares the device SIM level against the
5		.cei file certification key, and either allows the
		installation to proceed or prohibits it based on the
. 6		outcome of this comparison." (D, pg. 2)
7	,	"Security:. To achieve a high level of reliability,
8		WCEfA is carefully designed to: Control the installation of certified and tested
8	· •	software and drivers.
9		- Limit the access of system services by installed
		module.
10		- Monitor the proper execution of software"
.,		(G, pg. 1)
11.		The second load module is the certified software from
12	(d) distributing the second load module for use	the OEM or Microsoft that will be run as part of the
	by at least one device in the second device	"Closed" second device class.
13	class.	
14		"You may need to change ROM components after your
14		device ships, either to fix a problem, or to provide enhanced functionality. For this purpose, the OEM is
15	·.	given a CEIBuild that adds a master key to a .cei file.
		CEI files stamped with this master key can be installed
16		on an open, closed or a restricted platform." (D, pg. 3)
17		"Trusted: The application is registered as a completely
17		trusted module and allowed full access to the kernel
18		APIs and WCEfA APIs. This mode is mostly likely
·		reserved for MS and OEM applications and drivers.
19		Note that applications and drivers included in ROM are
20		automatically given trusted status." (I, pg.1)
20		L
21	References: [D] http://msdn.microsoft.com/library/default.asp?url=/lib	prary/en-us/duceauto/html/WinCAuto_SIM.asp
	rr://medn microsoft com/library/default.asp?url=/lib	rary/en-us/apcguide/nim/ceibuildrev_o.asp
22	tor 1 // made missocoft com/library/default asn?url=/lib	orary/en-us/apcguide/htm/securityrev.asp
	tara new //d= misrosoft com/library/default asn?url=/lib	rary/en-us/adcguide/htm/securityrev /.asp
23	crass = t/d= ==i-crase0 com/library/default asn?url=/libr	arv/en-us/apcguige/htm/feliabilityfev_3.asp
24	thu beautimed microsoft com/library/default asn?url=/lib	orary/en-us/wcedsn4u/num/cgconveniyingoignature.asp
<b>∠</b> ¬	[O] http://msdn.microsoft.com/library/default.asp?url=/lib	Dental According Human Commonsh
25		
]		
26		
27		•
. 21		_
28		
	···	

2.	FOR U.S. PATENT NO. 6,157,721	
3		
4	5.	Product infringing: Windows Hardware Quality Lab certification services, and
5		operating system products that support
6	A software verifying method comprising:	driver signature technology.  Microsoft encourages manufacturers to
7	·	have their device drivers tested and signed. For example, only signed drivers will ship
8		"in-the-box." Also, Microsoft's driver ranking prefers signed drivers to unsigned
9		drivers.
10		Microsoft Web Page - Can't Find a Test Category for Your Driver?
11		WHQL's long-term objective is to be able to digitally sign all drivers. Although we do
12		not currently have test programs for certain driver types, such as specialized device
13		drivers and software filter drivers, WHQL is investigating a long term solution to
14		expand the categories of drivers tested under Windows 2000 and ultimately all
15		Windows operating systems. We are already formulating a test program for anti-
16		virus file system filters, and plan to address other file system filter drivers as soon as
17	(a) testing a load module	the initial program is in place.  The driver will be tested for each version of
18		the operating system it supports and against the device class specification that apply to
19		the device's class.
20		The driver package is a load module. A driver package contains one or more of the
21.		following files: A device setup information file (INF file)
22		A driver catalog (.cat) file One or more optional co-installers
23		Microsoft operates the Window Hardware
24		Quality Lab, which tests drivers submitted by driver manufactures.
25	,	The manufacturer can test their own driver
26		using the Microsoft testing kit and submit the test results to WHQL when requesting a
27		signature. Additionally, Microsoft or a testing facility working with Microsoft can
28	having at least one specification associated	perform the testing.  The manufacturer-written INF file, which
	Daving at least one specification associated	

therewith,  is part of the driver package, specification. Microsoft Wir must have an INF file in order installed.	ndows drivers
must have an INF file in order installed.	er to be
li instance.	
the specification describing one or more  The INF Version section specific design class. One use of the	cifies its
functions performed by the load module; device class. One use of the	device class is lows
compatibility specification to	hat relate to the   ations will vary
by device class in part becau	ong class. The
INF incorporates by reference Microsoft supplied device cl	ce the lass-specific
specification by identifying in INF.	its class in the
The INF can include operating decorating to specify the constraint and the system architecture, major and the system architecture, major and the system architecture.	operating and minor
version, product and suite th intended for and can further	ne driver is use this
decorating to specify what o systems for which it is not in	ntended.
Because the functionality of operating systems may vary	the driver must [
be tested for each applicable system.	s operating
Oualification Service Policy Hardware Category Policies	y Guide – s
You must select the correct	
category for your device. If wrong hardware category fo	you select the or your device,
your submission will fail. For you have a storage/hard driver	or example, ii ve device, but
you select storage/tape drive hardware category, your sul	e as your
fail.	•
Windows XP HCT 10.0 Q o XP Logos	& A – Windows
22	Jindows XP"
Q: Which "Designed for W logos are available for my p	product?
A: Devices and systems qualified by the control of	ogo after passing
testing with the appropriate on all operating systems sp	ecitied by the
logo. "Designed for Windows" and System Programs lists whi	Logos for Device ich logos are
available for each type of p	product.
(b) verifying that the load module satisfies the specification; and (b) verifying that the load module satisfies (Compatibility Test (HCT)) includes the tests, test documents.	kit version 10.0
microres the tests, test does	

1 2 3 4 5 6 7 8		submission processes that are required to participate in the Microsoft Windows Logo Program for Hardware for the Windows XP Professional operating system. To qualify to use the "Designed for Windows" logo for hardware, products must pass testing with the Microsoft Windows HCT kit. The HCT kits are organized by hardware type.  As mentioned above, the manufacturer can test their own driver using the Microsoft testing kit and submit the test results to WHQL when requesting a signature. Additionally, Microsoft or a testing facility working with Microsoft can perform the testing.
10	(c) issuing at least one digital certificate	When a driver package passes WHQL testing, WHQL generates a separate CAT
11	attesting to the results of the verifying step.	file containing a hash of the driver binaries
		and other relevant information. WHQL then digitally signs the CAT file using
12		Digital Signature cryptographic technology and sends it to the vendor. Driver signing
. 13		does not change the driver binaries or the INF file submitted for testing.
14		Microsoft uses digital signatures for device
15		drivers to let users know that drivers are compatible with Microsoft Windows XP,
16		Windows 2000, and Windows Me. A
17		driver's digital signature indicates that the driver was tested with Windows for
18		compatibility and has not been altered since testing.
19		<i>₹</i>
20		
21		
	·	
22		
24		
25		
26	·	
27		
28		
•	1	

اۃ		THE PARTY OF
3	CLAIMLANGUAGE	CLAIM OF INFRINGEMENT
4	14.	Infringing products include Office 2003 and
5		included applications, and Server 2003, including Microsoft hosted RMS Service using
6		Passport
7	A first protected processing environment	A personal computer running Windows XP, Windows 2000, or Windows 2003
8	comprising: a first tamper resistant barrier having a first	The tamper resistant barrier is the Office 2003 IRM client environment and includes the
	security level, and	signed digital certificate identifying the user.
9		If the certificate is tampered with, or if certain,
10		sensitive IRM processes or modules are debugged or tampered with, the system will
11		cease to operate.
12		The first security level is the "Security Level" which has been selected for a particular Office
13		Application, e.g., Word.
14	at least one arrangement within the first tamper resistant barrier that prevents the first	The arrangement that prevents a load module from running in one PPE and not in another is
15	protected processing environment from executing the same load module accessed by a	the type and characteristics of a particular Load Module (VBA program within a document or
	executing the same rough anyment	add-in); i.e., signed, script author, code
16	second protected processing environment	capabilities, etc., and the "Security Level"
16 17	having a second tamper resistant barrier with a second security level different from the first	capabilities, etc., and the "Security Level" settings.
17	Il having a second tamper resistant parner with a	capabilities, etc., and the "Security Level"
17 18	having a second tamper resistant barrier with a second security level different from the first	capabilities, etc., and the "Security Level"
17 18 19	having a second tamper resistant barrier with a second security level different from the first	capabilities, etc., and the "Security Level"
17 18	having a second tamper resistant barrier with a second security level different from the first	capabilities, etc., and the "Security Level"
17 18 19	having a second tamper resistant barrier with a second security level different from the first	capabilities, etc., and the "Security Level"
17 18 19 20	having a second tamper resistant barrier with a second security level different from the first	capabilities, etc., and the "Security Level"
17 18 19 20 21	having a second tamper resistant barrier with a second security level different from the first	capabilities, etc., and the "Security Level"
17 18 19 20 21 22 23	having a second tamper resistant barrier with a second security level different from the first	capabilities, etc., and the "Security Level"
17 18 19 20 21 22 23 24	having a second tamper resistant barrier with a second security level different from the first	capabilities, etc., and the "Security Level"
17 18 19 20 21 22 23 24 25	having a second tamper resistant barrier with a second security level different from the first	capabilities, etc., and the "Security Level"
17 18 19 20 21 22 23 24 25 26	having a second tamper resistant barrier with a second security level different from the first	capabilities, etc., and the "Security Level"
17 18 19 20 21 22 23 24 25 26 27	having a second tamper resistant barrier with a second security level different from the first	capabilities, etc., and the "Security Level"
17 18 19 20 21 22 23 24 25 26	having a second tamper resistant barrier with a second security level different from the first	capabilities, etc., and the "Security Level"

Exhibit B

3	CLAIM LANGUAGE 1344 St	CLAIM OF INFRINGEMENT:
4	18.	Infringing products include Office 2003 and
5		included applications, and Server 2003, including Microsoft hosted RMS Service using
6		Passport
7	A method for protecting a first computing arrangement surrounded by a first tamper	The first computing arrangement with a tamper resistant barrier is the Office 2003 IRM client environment and includes the signed digital
8	resistant barrier having a first security level, the method including:	certificate identifying the user.
9		If the certificate is tampered with, or if certain, sensitive IRM processes or modules are
10		debugged or tampered with, the system will cease to operate.
11		The computing arrangement is being protected
12	· .	from; for example, viruses and malicious code.
13		The first security level is the "Security Level" which has been selected for a particular Office
14	preventing the first computing arrangement	Application, e.g., Word.
15	from using the same software module accessible by a second computing arrangement	The arrangement that prevents a load module from running in one computing arrangement
16	having a second tamper resistant barrier with a second security level different from the first	and not in another is the type and characteristics of a particular software module
17	security level.	(VBA program within a document or add-in); i.e., signed, script author, code capabilities,
18	·	etc., and the "Security Level" settings.
19		
20		
21		
22		
23		
24		
25		
26		
27	·	
28		

Exhibit B

293482.02

2	FOR U.S. PATENT NO. 6,157,721	
3	CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
4 5	34.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using
6		Passport
7	A protected processing environment comprising:	A personal computer running Windows XP, Windows 2000, or Windows 2003
8 9 10	a first tamper resistant barrier having a first security level,	The first tamper resistant barrier is the Office 2003 IRM client environment and includes the signed digital certificate identifying the user. If the certificate is tampered with, or if certain, sensitive IRM processes or modules are debugged or tampered with, the system will cease to operate.
11		The first security level is the "Security Level" which has been selected for a particular Office Application, e.g., Word.
13 14 15	a first secure execution space, and	The secure execution space is process space allocated by the operating system for the Microsoft Office host application to run. This host application (e.g., Word) executes the VBA code within this process space.
16 17 18		This execution space (application) is secure because the IRM environment takes steps to insure that it is "trusted", the application is signed, and the document which includes the VBA code is protected by IRM policy and then encrypted and signed.
19 20	at least one arrangement within the first tamper resistant barrier that prevents the first	The arrangement that prevents a load module from running in one computing arrangement
21	secure execution space from executing the	and not in another is the type and
22	execution space having a second tamper resistant barrier with a second security level different from the first security level.	(VBA program within a document or add-in); i.e., signed, script author, code capabilities, etc., and the "Security Level" settings.
23 24		
25		

Exhibit E

26

27

28

FOR U.S. PATENT NO. 6,157,721	
3   STATES OF THE STATE OF THE	ASSESSION OF THE PROPERTY OF T
34.	Product Infringing: Microsoft Common Language Runtime and ASP.NET
A protected processing environmer comprising:	I ASP.NET
a first tamper resistant barrier havi first security level,	ing a TAMPER RESISTANT BARRIER The first tamper resistant barrier is the application domain in the CLR. The runtime hashes the
	domain and compares it with the hash value in the manifest. If two hashes don't match, the assembly
	fails to load.[1]
1.	Also "Code running in one application cannot directly access code or resources from another application. The common language runtime
2	enforces this isolation by preventing direct calls between objects in different application domains.
3.	Objects that pass between domains are either copied or accessed by proxy."[2]
	SECURITY LEVELS
5	The security levels of the application domain if different by setting the trust level assigned to an
7	outside application using the "trust" element in the web.config for the ASP.NET application.  Syntax-
3	<trust level="Full/High/Low/None" originurl="url"></trust>
	Example- <trust <="" level="High" td=""></trust>
	originUrl=http://www.SomeOtherCompany.com/default.aspx/>
	[7]
a first secure execution space, and	particular application.
at least one arrangement within the tamper resistant barrier that preven first secure execution space from	The second secure execution space is another application domain that has a different trust level fo an outside application.
first secure execution space from executing the same executable accord by a second secure execution space	essed If second app domain gives Full trust to the outside
having a second tamper resistant be with a second security level differen	arrier application; whereas the first one doesn't, the first app domain won't be able to execute the application
the first security level.	References:

1 2 3 4	www.microsoft.com/germany/ms/msdnbiblio/do tnetrk/doc/assembly.doc [2] msdn.Microsoft.com/library/en- us/cpguide/html/ cpconapplicationdomainsoverview.asp?frame=tr ue [7] LaMacchia,etc, NET Framework Security, Addision-Wesley, 2002
6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26	
27 <b>28</b>	

3	FOR U.S. PATE	N1 NU. 0,157,721
	I MANAGEMENT OF ATMITANCING GENERAL STATES	MANAGEAIM OF INFRINGEMENTS AND AND AND AND AND AND AND AND AND AND
4	34.	Product Infringing: Products containing
5		Microsoft Common Language Runtime or
		Compact Common Language Runtime and
6		products implementing the Common Language
		Infrastructure specification.  Microsoft Common Language Runtime and
7	A protected processing environment comprising:	.NET Framework SDK:
8	a first tamper resistant barrier having a first	TAMPER RESISTANT BARRIER
	security level,	The first tamper resistant barrier is the
9.		application domain in the CLR. The runtime
		hashes the contents of each file loaded into the application domain and compares it with the
10		hash value in the manifest. If two hashes don't
11	·	match, the assembly fails to load. [1]
**		
12		Also "Code running in one application cannot directly access code or resources from another
١,, ١		application. The common language runtime
13		enforces this isolation by preventing direct
14		calls between objects in different application
. 1	·	domains. Objects that pass between domains
15		are either copied or accessed by proxy."[2]
16		SECURITY LEVELS
	·	Application domains have different security
17		levels by setting security policy of the
18		application domain programmatically. [3]
		"It has different security based on code-based
19		security model of .NET. Administrators and
	`	hosts use code-access security to decide what code can do, based on characteristics of the
20		code itself, regardless of what user is executing
21		the code. The code characteristics are called
~ .		evidence and can include the Web site or zone
22	•	from which the code was downloaded, or the digital signature of the vendor who published
22		the code."
23		
24		"When the security manager needs to
	• • • • • • • • • • • • • • • • • • • •	determine the set of permissions that an
25		assembly is granted by security policy, it starts with the enterprise policy level. Supplying the
26		assembly evidence to this policy level will
26		result in the set of permissions granted from
27		that policy level. The security manager
	•	typically continues to collect the permission
28		sets of the policy levels below the enterprise policy [including the app domain] in the same
. }		1 portor, 1 and 1 portor 1 and 1 portor 1 and 1 portor 1
	· ·	

Exhibit B

1

1 2 3 4		fashion. These permission sets are then intersected to generate the policy system permission set for the assembly. All levels must allow a specific permission before it can make it into the granted permission set for the assembly."
5 6		Example of granted permission sets from a policy — Condition: All code, Permission Set: Nothing
7		Condition: Zone: Internet, Permission Set: Internet Condition: URL:
•		www.monash.edu.au, Permission Set: MonashPSet
8		Condition: Strong Name: m-Commerce, Permission Set: m-
· <b>9</b>		CommercePSet [4]
10 11		Another difference in security levels can be whether the verification process is turned off or
12		on, "Managed code must be passed through a verification process before it can be run (unless the administrator has granted
13		permission to skip the verification). The verification process determines whether the
14		code can attempt to access invalid memory addresses or perform some other action that
15		could cause the process in which it is running to fail to operate properly. Code that passes
16		the verification test is said to be type-safe. The ability to verify code as type-safe enables the
17		common language runtime to provide as great a level of isolation as the process boundary, at a much lower performance cost." [5]
18		u much lower perjormance cost. [3]
19	a first secure execution space, and	The application domain is the execution space for a particular application.
20	at least one arrangement within the first tamper	The second secure execution space is another application domain that has a different security
21	resistant barrier that prevents the first secure execution space from executing the same	policy than the first.
22	executable accessed by a second secure execution space having a second tamper	If second app domain's security policy doesn't
23	resistant barrier with a second security level different from the first security level.	give any permission to code from internet zone, but first app domain does, then the code
24		would run in first app domain and not in second.[6]
25		References:
26		www.microsoft.com/germany/ms/msdnbibl io/dotnetrk/doc/assembly.doc
27		[2] msdn.Microsoft.com/library/en- us/cpguide/html/
28	· ·	cpconapplicationdomainsoverview.asp?fra me=true

1 2 3 4 5 6			[3] LaMacchia, etc, .NET Framework Security, Addision-Wesley, 2002, p.113 [4] Watkins, Demien, "An Overview of Security in the .NET Framework", from MSDN Library, January 2002 [5] same as [2] [6] msdn.Microsoft.com/library/en- us/cpguide/html/ cpconapplicationdomainlevelsecuritypolicy .asp?frame=true
8			
9			
10			
11			
12	•	•	
13			
14			
.15			
16			•
17		•	
18			
19			•
20	·		
21			
22			
24		•	
- 1			
25 26			
1			
27 28			
. 20	·		and the second of the second of the second

293482.02

-	FOR U.S. PATENT NO. 0,157,721		
3	CLAIMLANGUAGE	CLAIM OF INFRINGEMENT	
.4	38.	Infringing products include Office 2003 and included applications, and Server 2003,	
5		including Microsoft hosted RMS Service using	
6.	A method for protecting a first computing	Passport The first computing arrangement surrounded by	
7	arrangement surrounded by a first tamper resistant barrier having a first security level,	a tamper resistant barrier is the Office 2003 IRM client environment and includes the	
. 8	the method including:	signed digital certificate identifying the user. If the certificate is tampered with, or if certain,	
9		sensitive IRM processes or modules are debugged or tampered with, the system will	
10		cease to operate.  The first security level is the "Security Level"	
11 12		which has been selected for a particular Office Application, e.g., Word.	
13	preventing the first computing arrangement from using the same software module accessed	The computing arrangement that prevents a	
14	by a second computing arrangement having a second tamper resistant barrier with a second	software module from running in one computing arrangement and not in another is	
15	security level different from the first security level.	the type and characteristics of the particular software module (VBA program within a	
16		document or add-in); i.e., signed, script author, code capabilities, etc., and the "Security Level"	
17		settings.	
18			
19			
20			
21			
22			
23			
24			
25		٠.	
26			
27			
28	·		
ì			

3		
	Z#Z#ZZZZGEATMIEANGUAGE#Z#ZZZZZ	SESSE CIAIMOFINERINGEMENTAL
4	2.	Product Infringing: Windows Media Rights Manager and Windows Media Player
5 ∦	A starting lading	
6	A system including: (a) a first apparatus including,	Consumer's computer, as shown in WMRM SDK
.7	(1) user controls,	Consumer's computer, as shown in WMRM SDK
8	(2) a communications port,	Consumer's computer, as shown in WMRM SDK
9	(3) a processor,	Consumer's computer, as shown in WMRM SDK
10	(4) a memory storing:	Consumer's computer, as shown in WMRM SDK
11	(i) a first secure container containing	Secure container (packaged Windows Media
	a governed item, the first secure	file), received by consumer's computer from "Content provider" (WMRM SDK, Step 3),
12	container governed item being at	which contains encrypted governed item
	least in part encrypted; the first	("Encrypted content")
13	secure container having been received from a second apparatus;	
14	(ii) a first secure container rule at least	Rights portion of signed license, received by
17	in part governing an aspect of	consumer's computer from "License issuer"
15	access to or use of said first secure	(WMRM SDK, Step 9)
	container governed item, the first	
16	secure container rule [sic], the first	
	secure container rule having been received from a third apparatus	
17	different from said second	
18	apparatus; and	
10	(5) hardware or software used for	Windows Media Player and Windows Media
19	receiving and opening secure	Rights Manager
	containers, said secure containers each	
20	including the capacity to contain a	
	governed item, a secure container rule	
21	being associated with each of said	
22	secure containers; (6) a protected processing environment at	1st and 2nd rules consist of any two valid rules
22	least in part protecting information	as specified in the Window Media Rights
23	contained in said protected processing	Manager SDK: protected processing
ر دید	environment from tampering by a user	environment includes Windows Media Rights
24	of said first apparatus, said protected	Manager and Windows processes for
	processing environment including	protecting operation of Windows Media Rights
25	hardware or software used for	Manager. Licenses can be used to convey
	applying said first secure container	multiple rules.
26	rule and a second secure container rule	·
	in combination to at least in part	
27	govern at least one aspect of access to	
20	or use of a governed item contained in	•
28	a secure container; and (7) hardware or software used for	Any hardware or software employed in
	(/) nardware of Software used for	

	transmission of secure containers to other apparatuses or for the receipt of secure containers from other apparatuses.	transmitting Windows Media files, including for example consumer's computer's communication port and Windows Media Player (WMRM SDK, Step 3)
- <b>-</b>		
	45.1	
	•	
•		
-		
	· .	
•	•	
	•	

202462.02

#### INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP. INTERTRUST INFRINGEMENT CHART

2	INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 6,185,683		
3	CLAIM LANGUAGE	CLAIM OF INFRINGEMENT	
5 6	2.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport	
*	A system including:		
7	a first apparatus including,	A device with user controls, a communications	
8	user controls,	port, a processor and memory. For example, the user controls may be a keyboard and	
9	a communications port,	mouse, the communications port may be a NIC card with an Ethernet port, the processor may be a CPU, and the memory may be a hard-drive	
. 10	a processor,	or RAM.	
11	a memory storing:		
12	a first secure container containing a governed item, the first secure container governed item	An encrypted IRM-governed email received from a remote computer. The encrypted IRM-	
13	being at least in part encrypted; the first secure container having been received from a second	governed email contains an encrypted IRM- governed email message.	
14	apparatus;		
15	a first secure container rule at least in part governing an aspect of access to or use of said	The first secure container rule is received from the RMS server in the form of a use license.	
16	first secure container governed item, the first secure container rule, the first secure container	This use license contains rules generated by the	
17	rule having been received from a third apparatus different from said second apparatus; and	RMS server specifically for the user (or user's group)	
18	hardware or software used for receiving and	The RM-enabled device contains hardware or	
19	opening secure containers,	software for receiving and opening secure emails.	
20	said secure containers each including the capacity to contain a governed item, a secure	The secure email has the capacity to contain an	
21	container rule being associated with each of said secure containers;	IRM-governed email message, with a rule being associated with each email.	
22		The rules associated with the secure emails are	
23		rules that come as part of the original email as well as rules that come back from the RMS.	
24	a protected processing environment at least in part protecting information contained in said	Protected information on the RM-enabled device is protected by the use of at least	
25	protected processing environment from tampering by a user of said first apparatus,	cryptographic techniques.	
26	sold protected processing environment	The rule governing the email works together	
27	said protected processing environment including hardware or software used for applying said first secure container rule and a	with an additional rule to determine what access to or use (if any) are allowed with	
28	second secure container rule in combination to	respect to the IRM-governed email message.	

Exhibit B

respect to the IRM-governed email message.

For example, the additional rule may be

28

293482.02

second secure container rule in combination to

at least in nart govern at least one asnect of

access to in a secur	or use of a governed item e container; and	contained	received together with the rule in the use license.
hardware secure co the receip apparatus	or software used for transition trainers to other apparatult of secure containers from the secure cont	smission of ses or for m other	The device includes hardware or software used for transmitting or receiving secure emails. For example, RM-enabled OUTLOOK is designed to transmit and receive encrypted IRM-governed emails to/from other devices.
		•	
	•		
		·	
·			
	•		
	·.		•
		,	
	·		n

293482.02

3		
4	CLAIM LANGUAGE SEL	CLAIMOF INERINGEMENT
5	2.	Infringing products include Office 2003 and included applications, and Server 2003,
6		including Microsoft hosted RMS Service using Passport
7	A system including:	
8	a first apparatus including,	A device with user controls, a communications port, a processor, and memory. For example,
9	user controls,	the user controls may be a keyboard and mouse, the communications port may be a NIC
10	a communications port,	card with an Ethernet port, the processor may be a CPU, and the memory may be a hard-drive
11	a processor,	or RAM.
12	a memory storing: a first secure container containing a governed	The first secure container is an encrypted IRM-
.13	item, the first secure container governed item	protected document.
14	container having been received from a second apparatus;	This encrypted IRM-governed document is, for example, received from a remote computer, as an attachment to an IRM-governed email or
15 16		downloaded from a document server or web site.
17	a first secure container rule at least in part	The first secure container rule is received from the RMS server in the form of a use license.
18	governing an aspect of access to or use of said first secure container governed item, the first secure container rule, the first secure container	This use license contains rules generated by the
19	rule having been received from a unit apparatus different from said second	RMS server specifically for the user (or user's group).
20	apparatus; and hardware or software used for receiving and	The RM-enabled device contains hardware or
21	opening secure containers,	software for receiving and opening secure documents.
22	said secure containers each including the capacity to contain a governed item, a secure	The secure documents have the capacity to
23	container rule being associated with each of said secure containers;	contain IRM-governed content, with a rule being associated with each secure document.
24		The rules associated with said secure
25		documents are the rules that come as part of the originally received document as well as rules that come back from the RMS server.
26	a protected processing environment at least in	Protected information on the RM-enabled
27	nari projecting information contained in Salu	device is protected by the use of at least cryptographic technique.
28	protected processing environment from tampering by a user of said first apparatus,	The rule governing the document works
		The rule governing the document works

including hardware or software used for applying said first secure container rule and a second secure container rule in combination to at least in part govern at least one aspect of access to or use of a governed item contained in a secure container; and	together with an additional rule to determine what access to or use (if any) are allowed wit respect to the IRM-governed document. For example, the additional rule may be associate with an email to which the document was attached, or received together with the rule in the use license.
hardware or software used for transmission of secure containers to other apparatuses or for the receipt of secure containers from other apparatuses.	The device includes hardware or software use for transmitting or receiving secure documents For example, RM-enabled OUTLOOK is designed to transmit and receive to/from othe devices emails with IRM-governed document attached thereto.
	<u>-2.</u>
•	·
·	
	•.
•	
	· .
•	

, 3		
4	CLAIM LANGUAGE	CLAIM OF INERINGEMENTALES
5	<b>3.</b>	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using
6	<b>}</b>	Passport
7	A system including:	
0	a first apparatus including,	A device with user controls, a communications port, a processor and memory. For example,
8	user controls,	the user controls may be a keyboard and
9	a communications port,	mouse, the communications port may be a NIC card with an Ethernet port, the processor may
10	a processor,	be a CPU, and the memory may be a hard-drive or RAM.
11	a memory storing:	
12	a first secure container containing a governed item, the first secure container governed item	The first secure container containing a governed item is an IRM protected email.
13	being at least in part encrypted;	Both the email and attachment are IRM
14		protected, each having their own rules, each being encrypted.
15	a first secure container rule at least in part governing an aspect of access to or use of said	The rule governing the email (a first secure container rule) governs said first secure
16	first secure container governed item; and	container governed item.
.17	\	
18	a second secure container containing a digital certificate;	The second secure container is the IRM protected attachment's derived license request
19		object. The license request object contains the
20		Publishing license and a signed digital certificate.
		conneate.
21	hardware or software used for receiving and	The DM (IDM) and the state of the Company of the Co
22	opening secure containers,	The RM (IRM) enabled computer has software for receiving and opening secure containers.
23	said secure containers each including the	The IRM secure containers have capacity to
24	capacity to contain a governed item, a secure container rule being associated with each of	contain a governed item, with a secure container rule being associated with each of
25	said secure containers;	said secure containers.
	a protected processing environment at least in part protecting information contained in said	Protected information on the RM-enabled computer is protected by the use of at least
26	protected processing environment from tampering by a user of said first apparatus,	cryptographic techniques.
27		
28	said protected processing environment including hardware or software used for	The rules governing the email itself (first
Ì	The state of the s	The raise streething the chian fisen this.

Exhibit B

293482.02

applying said first secure container rule and a second secure container rule in combination to at least in part govern at least one aspect of access to or use of a governed item contained in a secure container; and	secure container rule) and the rules governing the attachment work together to determine wha access to or use (if any) will be allowed with respect to the governed item.
hardware or software used for transmission of secure containers to other apparatuses or for the receipt of secure containers from other apparatuses.	IRM-enabled applications, e.g., OUTLOOK, are designed to transmit and receive RM secured containers to/from other computers.
	-
· · · · · · · · · · · · · · · · · · ·	
	·

The second secon	
CLAIMLANGUAGE	CLAIM OF INFRINGEMENT
3.	Infringing products include Office 2003 and
	included applications, and Server 2003,
	including Microsoft hosted RMS Service usin
·	Passport
A system including:	
a first apparatus including,	A device with user controls, a communication
user controls,	port, a processor and memory. For example, the user controls may be a keyboard and
a communications port,	mouse, the communications port may be a NI card with an Ethernet port, the processor may
a processor,	be a CPU, and the memory may be a hard-dri or RAM.
a memory storing:	
a first secure container containing a governed	The first secure container containing a
item, the first secure container governed item	governed item is an IRM protected document,
being at least in part encrypted;	which is an attachment within an IRM
	protected email message. The governed item in the document's content.
	Both the email message and attachment are
	encrypted and have associated usage rules due to IRM protection.
a first secure container rule at least in part	A use license for the IRM protected documen
governing an aspect of access to or use of said	specifies rules governing access to or use of
first secure container governed item; and	said first secure container governed item.
a second secure container containing a digital	The second secure container is the IRM
certificate;	protected email message.
	The IRM protected attachment includes a
	publishing license and an owner certificate,
	both of which are signed XrML digital
	certificates.
	The attachment (including embedded
	certificates) is contained within the IRM
	protected email message (said second secure
·	container).
hardware or software used for receiving and	The RM (IRM) enabled computer has softwar
opening secure containers,	for receiving and opening secure containers.
and a sum containers and including the	The IDM course containers have compainted
said secure containers each including the capacity to contain a governed item, a secure	The IRM secure containers have capacity to contain a governed item, with a secure
container rule being associated with each of	container rule being associated with each of
said secure containers:	said secure containers.
a protected processing environment at least in	Protected information on the RM-enabled
part protecting information contained in said	computer is protected by the use of at least
protected processing environment from	cryptographic techniques.
	ii .
	•
Exhi	ibit B
II	ibit B

293482:02

1

1	tampering by a user of said first apparatus,	
2	said protected processing environment	m
3	including hardware or software used for applying said first secure container rule and a	The rules governing the attachment (first secure container rule) and the rules governing the
4	second secure container rule in combination to at least in part govern at least one aspect of	email message (second secure container rule) work together to determine what access to or
5	access to or use of a governed nem contained	use (if any) will be allowed with respect to the
6	in a secure container; and hardware or software used for transmission of	RM-enabled applications, e.g., OUTLOOK, are designed to transmit and receive RM secured
7	secure containers to other apparatuses or for the receipt of secure containers from other	containers to/from other computers.
	apparatuses.	
8	4. A system as in claim 3,	All parts of the attachment (including
. 9	said memory storing a rule associated with said second secure container, said rule	embedded signed XrML licenses/certificates)
10	associated with said second secure container at least in part governing at least one aspect of	and governed by the associated email rules (second secure container rule).
11	access to or use of said digital certificate.	1 (Second Secure container raish
12		
13		
14		
. 15		
16		
17		
18		•
19		
20	·	
21		
22		
23	·	•
24		
25		
26		•
27		
28		

٠.	FOR U.S. PATE	141 140. 0,100,000				
3	CLAIM LANGUAGE	CLAIM OF INFRINGEMENT				
. 4	5.	Infringing products include Office 2003 and				
5		included applications, and Server 2003, including Microsoft hosted RMS Service using				
6		Passport				
7	A system including: a first apparatus including,	A device with user controls, a communications				
8	user controls,	port, a processor and memory. For example, the user controls may be a keyboard and				
9	a communications port,	mouse, the communications port may be a NIC card with an Ethernet port, the processor may				
10	a processor,	be a CPU, and the memory may be a hard-drive or RAM.				
11	a memory storing:					
12	a first secure container containing a governed item, the first secure container governed item	first secure container containing a governed item is an IRM protected email.				
13	being at least in part encrypted;	Both the email and attachment are IRM protected, each having their own rules, each				
14		being encrypted.				
15 16	a first secure container rule at least in part governing an aspect of access to or use of said	The rule governing the email (a first secure container rule) governs said first secure container governed item.				
17	first secure container governed item; and	container governed nem.				
18	a second secure container containing a digital signature, the second secure container being	The second secure container is the IRM protected attachment's derived license request				
19	different from said first secure container;	object. The license request object contains the				
20		Publishing license and a signed digital certificate.				
21	hardware or software used for receiving and	The RM (IRM) enabled computer has software				
22	opening secure containers, said secure containers each including the capacity to	for receiving and opening secure containers.				
23	contain a governed item, a secure container rule being associated with each of said secure	The IRM secure containers have capacity to contain a governed item, with a secure container rule being associated with each of				
24	containers;	said secure containers.				
25	a protected processing environment at least in part protecting information contained in said	Protected information on the RM-enabled computer is protected by the use of at least				
26	protected processing environment from tampering by a user of said first apparatus,	cryptographic techniques.				
27	said protected processing environment					
28	including hardware or software used for applying said first secure container rule and a	The rules governing the email itself (first secure container rule) and the rules governing				
. [						

Exhibit B 73

293482.02

	second secure container rule in combination to at least in part govern at least one aspect of access to or use of a governed item contained in a secure container; and	the attachment will work together to determine what access to or use (if any) will be allowed with respect to the governed item.
	hardware or software used for transmission of secure containers to other apparatuses or for the receipt of secure containers from other apparatuses.	RM-enabled applications, e.g., OUTLOOK, are designed to transmit and receive RM secured containers to/from other computers.
ľ		
	•	
	•	
		•

3		
.4	CLAIM LANGUAGE	GLAIM OF INFRINGEMENT
5	5.	Infringing products include Office 2003 and included applications, and Server 2003,
6		including Microsoft hosted RMS Service using Passport
7	A system including:	1 dasport
8	a first apparatus including,	A device with user controls, a communications port, a processor and memory. For example,
9	user controls,	the user controls may be a keyboard and mouse, the communications port may be a NIC
10	a communications port,	card with an Ethernet port, the processor may be a CPU, and the memory may be a hard-drive
11	a processor,	or RAM.
12	a memory storing:	first secure container containing a governed
13	a first secure container containing a governed item, the first secure container governed item being at least in part encrypted;	item is an IRM protected email.
14	being at least in part enerypiece,	Both the email and attachment are IRM protected, each having their own rules, each being encrypted.
15	a first secure container rule at least in part	The rule governing the email (a first secure
16	governing an aspect of access to or use of said first secure container governed item; and	container rule) governs said first secure container governed item.
17		·
18	a second secure container containing a digital signature, the second secure container being	The second secure container is the IRM email attachment.
19	different from said first secure container;	This attachment and its publishing license are
20		signed.
21	hardware or software used for receiving and opening secure containers, said secure	The RM (IRM) enabled computer has software for receiving and opening secure containers.
22	containers each including the capacity to contain a governed item, a secure container	The IRM secure containers have capacity to
23	rule being associated with each of said secure containers;	contain a governed item, with a secure container rule being associated with each of
24		said secure containers.
25	a protected processing environment at least in part protecting information contained in said protected processing environment from	Protected information on the RM-enabled computer is protected by the use of at least cryptographic techniques.
26	tampering by a user of said first apparatus,	,
27	said protected processing environment	The rules governing the email itself (first
28	including hardware or software used for applying said first secure container rule and a	secure container rule) and the rules governing

Exhibit B 75

293482.02

ع. 3	second secure container rule in combination to at least in part govern at least one aspect of access to or use of a governed item contained in a secure container; and	the attachment work together to determine what access to or use (if any) will be allowed with respect to the governed item.
l s t	nardware or software used for transmission of ecure containers to other apparatuses or for the receipt of secure containers from other	RM-enabled applications, e.g., OUTLOOK, are designed to transmit and receive RM secured containers to/from other computers.
a	pparatuses.	
	·	
ŀ		
	•	•
		•
	•	
		· .
		•
		•
	·	•
	•	

-		•
4	CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
5		Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
7	A system including:	1 assport
. 8	a first apparatus including,	A device with user controls, a communications
9	user controls,	port, a processor and memory. For example, the user controls may be a keyboard and mouse, the communications port may be a NIC
10	a communications port,	be a CPU, and the memory may be a hard-drive
11	a processor, a memory storing:	or RAM.
12	a first secure container containing a governed	The first secure container containing a
13	item, the first secure container governed item being at least in part encrypted;	governed item is an IRM protected document, which is an attachment within an IRM
14		protected email message. The governed item is the document's content.
15		Both the email message and attachment are
16		encrypted and have associated usage rules due to IRM protection.
17 18	a first secure container rule at least in part governing an aspect of access to or use of said first secure container governed item; and	A use license for the IRM protected document specifies rules governing access to or use of said first secure container governed item.
19	a second secure container containing a digital signature, the second secure container being different from said first secure container;	The second secure container is the IRM protected email message.
20	and the sound container,	The IRM protected attachment includes a
21		publishing license and an owner certificate, both of which are signed XrML digital certificates.
22	·	
- 23		The attachment (including embedded certificates) is contained within the IRM protected email message (said second secure
.24	hardware or software used for receiving and	container).
25	opening secure containers, said secure containers each including the capacity to	The RM (IRM) enabled computer has software for receiving and opening secure containers.
26	contain a governed item, a secure container rule being associated with each of said secure	The IRM secure containers have capacity to contain a governed item, with a secure
27	containers;	container rule being associated with each of said secure containers.
28	a protected processing environment at least in nart protecting information contained in said	Protected information on the RM-enabled commuter is protected by the use of at least
- 1		1."

Exhibit B

293482.02

1

	)	
2	protected processing environment from tampering by a user of said first apparatus,	cryptographic techniques.
3	said protected processing environment including hardware or software used for	The rules governing the attachment (first secure container rule) and the rules governing the
4	applying said first secure container rule and a second secure container rule in combination to	email message (second secure container rule)
5	at least in part govern at least one aspect of access to or use of a governed item contained	work together to determine what access to or use (if any) will be allowed with respect to the governed item.
6	in a secure container; and hardware or software used for transmission of	RM-enabled applications, e.g., OUTLOOK, are
,	secure containers to other apparatuses or for the receipt of secure containers from other	designed to transmit and receive RM secured containers to/from other computers.
8	apparatuses.	
	6. A system as in claim 5,	All parts of the attachment (including
	said memory storing a rule at least in part governing an aspect of access to or use of said digital signature.	embedded signed XrML licenses/certificates)
		and governed by the associated email rules (second secure container rule).
	·	
	·	·
		•
; ;		
7		
28		

3		
4	CLAIM EANGUAGE	CLAIM OF INFRINGEMENT
. 5	28.	Infringing products include Office 2003 and included applications, and Server 2003.
6		including Microsoft hosted RMS Service using Passport
·7	A system including:	rassport
. 8	a first apparatus including;	A device with user controls, a communications
. 9	user controls,	port, a processor and memory. For example, the user controls may be a keyboard and
10	a communications port,	mouse, the communications port may be a NIC card with an Ethernet port, the processor may be a CPU, and the memory may be a hard-drive
-11	a processor,	or RAM.
12	a memory containing a first rule,	The first rule governs use of an IRM protected document (e.g., an IRM rule permitting a
13		document to be read by specified users or barring access to IRM-governed information
14		from specified users, applications, or other principals).
15	hardware or software used for receiving and opening secure containers,	The RM-enabled device contains hardware or software for receiving and opening secure
16	said secure containers each including the	containers.
17	capacity to contain a governed item, a secure container rule being associated with each of	The secure email has the capacity to contain an IRM-governed email message, with a rule
18	said secure containers; a protected processing environment at least in	being associated with each email.  Protected information on the RM-enabled
19	part protected processing environment at least in part protecting information contained in said protected processing environment from	device is protected by the use of at least cryptographic techniques.
20	tampering by a user of said first apparatus,	
21	said protected processing environment including hardware or software used for	The secure container rule is an IRM rule governing access to the IRM protected
22	applying said first rule and a secure container rule in combination to at least in part govern at	document (e.g., a rule permitting editing by specified users).
23	least one aspect of access to or use of a governed item; and	The rule governing the email works together with an additional rule to determine what
24		access to or use (if any) are allowed with
25	-	respect to the IRM-governed email message (the document's content). For example, the
26		additional rule may be received together with the rule in the use license, may be associated with a publishing license, may be associated
27		with user certification, revocation lists, or exclusion policies, or may be received from
28		any other source.
}	hardware or software used for transmission of	The device includes hardware or software used

Exhibit B 79

293482.02

I

		•
. 1	coours containers to other apparetuses or for	Control time or receiving
	secure containers to other apparatuses or for the receipt of secure containers from other	for transmitting or receiving secure containers. For example, RM-enabled OUTLOOK is
2	apparatuses; and	designed to transmit and receive encrypted
3	apparatuses, and	IRM-governed emails to/from other devices.
٠ ع	a second apparatus including:	
. 4	user controls,	A device with user controls, a communications
•		port, a processor and memory. For example,
5	a communications port,	the user controls may be a keyboard and
		mouse, the communications port may be a NIC
6	a processor,	card with an Ethernet port, the processor may
_		be a CPU, and the memory may be a hard-drive
7	a memory containing a second rule,	or RAM.
8		The second rule coverns use of an IDM
0	·	The second rule governs use of an IRM protected document (e.g., an IRM rule
9		permitting a document to be read by specified
		users or barring access to IRM-governed
10		information from specified users, applications,
		or other principals).
11	hardware or software used for receiving and	The RM-enabled device contains hardware or
10	opening secure containers,	software for receiving and opening secure
12		containers.
13	said secure containers each including the capacity to contain a governed item, a secure	The secure email has the capacity to contain an
10	container rule being associated with each of	IRM-governed email item, with a rule being
14	said secure containers;	associated with each secure containers.
	a protected processing environment at least in	Protected information on the RM-enabled
15	part protecting information contained in said	device is protected by the use of at least
16	protected processing environment from	cryptographic technique.
16	tampering by a user of said apparatus,	
17	said protected processing environment	The secure container rule is an IRM rule governing access to the IRM protected
	including hardware or software used for	document (e.g., a rule permitting editing by
18	applying said second rule and a secure	specified users).
	container rule in combination to at least in part	,
19	govern at least one aspect of access to or use	The rule governing the email works together
20	of a governed item;	with an additional rule to determine what
20		access to or use (if any) are allowed with
21		respect to the IRM-governed item (the
	,	document's content). For example, the additional rule may be received together with
22		the rule in the use license, may be associated
		with a publishing license, may be associated
23	·	with user certification, revocation lists, or
ا ب		exclusion policies, or may be received from
24		any other source.
25	hardware or software used for transmission of	The device includes hardware or software used
ر ع	secure containers to other apparatuses or for	for transmitting or receiving secure containers.
26	the receipt of secure containers from other	For example, RM-enabled OUTLOOK is
	apparatuses; and	designed to transmit and receive encrypted IRM-governed emails to/from other devices.
27	an electronic intermediary, said intermediary	
	including a user rights authority clearinghouse.	The RMS Server (Microsoft hosted or otherwise) constructs a 'use license' specific to
28	mendanig a user rights authority clearinghouse.	a piece content and targets it to a specific user.
H		a piece comen and angers it to a specific user.
11	•	

1		
2	29. A system as in claim 28, said user rights authority clearinghouse operatively connected to make rights available to users.	The RMS server sends use licenses to users through a communications port, e.g., Ethernet, serial, satellite, "the internet" These use licenses include rights.
3		These use licenses include rights.
. 4		The clearing functionality of the RMS is operatively connected to the RMS server.
5		
6		
7		
8		
9		
٠,		
10		
11		
12		
13		
14		
15		
16		
17		
18		, <del>-</del> .
19		
20		
21		
22		
1		
23		•
24		
25	·	
į,		
26		
27		
28		
28		

293482.02

₩	<u> </u>	
28.		Product Infringing: Windows Media Rights Manager and Windows Media Player
A syst	em including:	
(a) a 1	first apparatus including;	Consumer's computer, as shown in WMRM SDK
(1)	) user controls,	Consumer's computer, as shown in WMRM SDK
(2	a communications port,	Consumer's computer, as shown in WMRM SDK
(3)	a processor,	Consumer's computer, as shown in WMRM SDK
(4)	a memory containing a first rule,	Memory is in the consumer's computer, first rule is a right received as part of a signed license (WMRM SDK, Step 9)
(5)	hardware or software used for receiving and opening secure containers, said secure containers each including the capacity to contain a governed item, a secure container rule being associated with each of said secure containers;	Consumer's computer receives Windows Media file (secure container) via communications port (WMRM SDK, Step 3) and applies secure container rule or rules via Windows Media Player and Windows Media Rights Manager.
(6)	a protected processing environment at least in part protecting information contained in said protected processing environment from tampering by a user of said first apparatus, said protected processing environment including hardware or software used for applying said first rule and a secure container rule in combination to at least in part govern at least one aspect of access to or use of a	Processing environment includes Windows Media Rights Manager and Windows processes for protecting operation of Windows Media Rights Manager
(7)	hardware or software used for transmission of secure containers to other apparatuses or for the receipt of secure containers from other	Hardware or software employed in transmitting Windows Media files, including for example consumer's computer's communication port and Windows Media Player (WMRM SDK,
43	apparatuses; and	Step 3)
	econd apparatus including:	2nd consumer's computer 2nd consumer's computer
(1)	user controls,	2nd consumer's computer  2nd consumer's computer
(2)	a communications port.	
(4)	a memory containing a second rule,	2nd consumer's computer  Memory is in the 2nd consumer's computer, first rule is a Right received as part of a signed license (WMRM SDK, Step 9)
(5)	hardware or software used for	2nd consumer's computer receives Windows
	receiving and opening secure containers, said secure containers each including the capacity to contain	Media file (secure container) via communications port (WMRM SDK, Step 3) and applies secure container rule or rules via

Exhibit B

293482.02

						• • • • • • • • • • • • • • • • • • • •				
1	<u> </u>			•			· 1: Di	1 37 27		. ,
2			a governed to rule being as said secure	ssociated v		Windows M Rights Man		and Win	dows Me	edia
		(6)	a protected pleast in part	processing protecting	environment at information	Processing of Media Righ	ts Manager	and Wind	lows	
		•	environmen	t from tam	cted processing pering by a said protected	processes for Media Right environment	ts Manager;	processi	ng	dow
			processing e hardware or	nvironmen software u	nt including used for	combination	1			•
	l L			iner rule in	ule and a n combination n at least one				-	
	· 		aspect of acc	cess to or u m;	se of a		•	•	• •	٠.
		(7)	hardware or transmission	of secure		Hardware of Windows M	ledia files, i	ncluding	for exam	ple
			secure containapparatuses;	iners from		port and Wi SDK, Step 3	ndows Med )			
	(c)	inter	lectronic intermediary includes	uding a use		License Issu	er			
-	29.		ority clearing stem as in cl							
П	said	lusei	rights author	rity clearin	ghouse	License Issu				- 0
	oper to us		ly connected	to make ri	ghts available	consumer's	computer (\	WMKM S	DK, Ste	р У)
ľ		<u> </u>								
			•			٠.,	•	•		
								•		
						•	•	· <u>-</u>	1	
						-				
					•					
			•				٠		•	
			. •							
							•			
					• <del>-</del>	*i	•••••			

3		
. 4	CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
5	56.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
7	A method of securely delivering an item, including the following steps:	
8	performing an authentication step;	The RM-enabled application, e.g., Word, OUTLOOK, PowerPoint, etc., must be authenticated before it is allowed access to or use of the content.
10	associating a digital signature with said item;	The RM protected content is signed.
11	incorporating said item into a first secure electronic container, said item being at least in part encrypted while in said container,	RM-protected content is packaged with rules and encrypted.
12	said incorporation occurring in an apparatus	
13	containing a first protected processing environment, said protected processing	Protected information on the RM enabled computer is protected by the use of at least
14	environment at least in part protecting information contained in said protected	cryptographic techniques.
15	processing environment from tampering by a user of said apparatus;	
16 17	in said protected processing environment, associating a first rule with said first secure electronic container, said first rule at least in	The IRM-protected document (said item) has an associated rule or rules.
18	part governing at least one aspect of access to or use of said item;	
19	authenticating an intended recipient of said item;	A recipient of IRM-protected content must be authenticated before being allowed access to or use of the content.
20 21	transmitting said first secure electronic container and said first rule to said intended recipient; and	The document is sent via IRM-protected email as an attachment.
22	using a second protected processing environment, providing said intended recipient	The email is received at another IRM-enabled computer.
23	access to at least a portion of said item,	
24	said access being governed at least in part by said first rule and by a second rule present at	The first said rule is the rule(s) associated with the attached document, and the second rule is
25	said intended recipient's site.	the rule(s) received that govern the email itself.
26		

Exhibit B 84

293482.02

27

28

1

4		
. 4	126.	Product Infringing: Windows Hardware
5		Quality Labs Authentication services,
		Windows operating Systems (such as Windows XP) that support the driver
6	·	signing features, and any product using
7		Driver Signing feature
•	A method of providing trusted intermediary	
8	services including the following steps: at a first apparatus, receiving an item from	Microsoft's Window Hardware Quality
ģ	a second apparatus;	Labs (WHQL) (first apparatus) receiving
9		driver package (item) from independent
10		hardware vendor (IHV) or any driver
	associating authentication information with	developer (second apparatus).  The signature information of a security
11	said item;	catalog file (see next element of claim)
12		names Microsoft as the publisher.
	·	WHQL's signature is intended to signify
13	<i>:</i>	that a driver has complied with Microsoft's Windows compatibility and/or Secure
14		Audio Path (SAP) specifications.
17	incorporating said item into a secure digital	The hashes of the files making up the
15	container;	driver package are included in the signed
16		security catalog file for the driver package.  The catalog file makes the driver package a
10		secure digital container.
17	associating a first rule with said secure	Driver developers specify rules in an INF
10	digital container, said first rule at least in part governing at least one aspect of access	file that govern the installation and/or use of the driver. For example, as specified in
18	to or use of said item;	the INF, the installation events will vary
19	· ·	based on the user's operating system
		version, which includes architecture, product type and suite. The INF logging
20		rules and can further specify security rules
21	•	that are evaluated when the driver is used.
		William Demonstration Co.
22	·	White Paper – Operating-System Versioning for Drivers under Windows XP
23		versioning for Differs and windows 74
		Setup selects the [Models] section to use
.24		based on the following rules:
25	•	If the INF contains [Models] sections for
23		several major or minor operating system
26		version numbers, Setup uses the section
l	-	with the highest version numbers that are
27		not higher than the operating system version on which the installation is taking
28		place.
20		
. 1		

Exhibit B

	1	
	2	
	3	
. •	4	
	5	
(	5	
7	7	
્રે 8	3	Ī
S	,	
10	. 11	
11	- 11	
12	- 11	
13		
14		
15		
16	1	
17		
18		
19		
20		
21		
23		
22		
23		•
24		
25		
26		
27		
	11	

If the INF [Models] sections that match the operating system version also include product type decorations, product suite decorations, or both, then Setup selects the section that most closely matches the running operating system.

Suppose, for example, Setup is running on Windows XP Professional (which is operating system version 5.1), and it finds the following entry in a [Manufacturer] section:

%FooCorp%=FooMfg, NT, NT.5, NT.5.5, NT....0x80

In this case, Setup will look for a [Models] section named [FooMfg.NT.5]. Setup will also use the [FooMfg.NT.5] section if it is running on a Datacenter version of Windows .NET Server, because a specific major/minor version takes precedence over the product type and suite mask.

For example, to create an INF that is intended for use only on Windows XP, the INF file could contain the following:

[Manufacturer]
"Foo Corp." = FooMfg, NT.5.1, NT.5.2
[FooMfg.NT.5.1]
"Foo Device" = FooDev. \*FOO1234

Note the omission of the undecorated [FooMfg] section, as well as the omission of the [FooMfg.NT.5.2] section. This INF file would appear to be "empty" on any operating system other than Windows XP.

## Access Control List Rules

XP DDK - Tightening File-Open
Security in a Device INF File
For Microsoft Windows 2000 and later,
Microsoft tightened file-open security in
the class installer INFs for certain device
classes, including CDROM, DiskDrive,
FDC, FloppyDisk, HDC, and
SCSIAdapter.

If you are unsure whether the class installer for your device has tightened security on file opens, you should tighten security by using the device's INF file to assign a value to the DeviceCharacteristics value name in the registry. Do this within an add-

		registry-section, which is specified using the INF AddReg directive.
2	transmitting said secure digital container	Microsoft, IHV, driver developer or any
3	and said first rule to a third apparatus, said	other party distributing signed driver
_	third apparatus including a protected	packages transmitting the driver package to
, 4	processing environment at least in part protecting information stored in said	user (third apparatus). Since the driver package includes the INF file, it will
5	protected processing environment from	include the first rule. The protected
J	tampering by a user of said third apparatus;	processing environment (PPE) is Windows
6		operating system with its pertinent services such as Windows File Protection, signature
· 7		and cryptographic functions, Plug and Play and Set-up and their related default and
8		modifiable policies. The PPE checks for
. 9		signatures on driver packages and detects situations when the driver package's
		signature does not match the driver
10		package.
11		Additionally, the Digital Rights Manager (DRM) components (kernel and client) will
12		contribute to making the third apparatus a
	1	PPE when the SAP functionality is
13		invoked. [That is, when SAP is required, an
1.4		additional signature is checked to verify that the driver is SAP compliant and that it
14	• •	hasn't been tampered with.]
15	said third apparatus receiving said secure	The end-user receiving the driver package.
	digital container and said first rule;	A star is the Diversity in the Control of the Contr
16	said third apparatus checking said authentication information; and	A step in the Plug and Play/Setup driver installation process checks signature at
17	dudionious monación, and	installation. Additionally, the DRM
• •		component will check the DRM signature
18	·	when invoking DRM functionality.
19		White Paper - Driver Signing for Windows
:20		During driver installation, Windows compares the hashes contained in the
21		driver's CAT file with the computed hash
	·	of the driver binaries to determine whether
22		the binaries have changed since the CAT file was created. If a driver fails the
23		signature check or there is no CAT file,
		what happens next depends on the driver
24		signing policy in effect on the user's system:
25		3,300111,
23		If the policy is set to Ignore, the driver
26		installs silently, with no message to the
		user.
27		If the policy is set to Warn, a message
28		warns the user the driver is unsigned,
	<u> </u>	which means that it has not passed WHQL

testing and might cause problems. The Warn dialog box gives an administrative user the option to override the warning and install an unsigned driver anyway. If the policy is set to Block, the system displays a message that informs the user that the driver cannot be installed because 5 it is not digitally signed. The action would be installing and/or using said third apparatus performing at least one 6 the driver. For example, installation action on said item, said at least one action policies govern the actions (ignore, warn or being governed, at least in part, by said 7 block) taken based on whether a driver is first rule and by a second rule resident at signed or not and these policies (rule) are said third apparatus prior to said receipt of resident on the third apparatus. Another said secure digital container and said first rule is the "ranking" of available drivers rule, said action governance occurring at least in part in said protected processing when selecting a driver to install. This ranking process includes whether a driver environment. 10 is signed or not. Another rule is the security access rules that the class installer 11 that will be used to install the device has. 12 In the case of DRM, the content will have associated rules governing its use in a SAP-13 complaint environment. These rules (the content license) can be resident at the third 14 apparatus particularly in the case when a user is installing a new (SAP-compliant) 15 device that will render previously acquired content or in the case that acquired content 16 cannot be rendered until the user installs required drivers. 17 For example, when installing: 18 The XP driver ranking process and the 19 modifiable default related to signature state of the driver act as the second rule. 20 The driver will be installed only if the first 21 and second rules validate. 22 Operating-System Versioning for Drivers under Windows XP 23 Default System Policy for Unsigned 24 Drivers 25 If the user installs an unsigned driver for a designated device class from disk or from 26 another web site, Windows XP/Windows 2000 displays a warning that the driver is 27 unsigned, thus helping to preserve the integrity of the released system. However, 28 by default, Windows XP/Windows 2000

does not block installation of unsigned drivers, so vendors can get urgent hot-fixes 2 to customers while waiting for WHQL to test the fix. In Windows XP, the default driver signing policy can be changed through the Hardware tab of the System applet on the Control Panel. A user can change the policy to be more restrictive, but not less restrictive on a per-user basis (that is, a user can change Warn to Block, but not to Ignore). An administrator can change the policy to be either more restrictive or less restrictive for all users on the system by checking "Apply the setting as system default. 10 Driver Ranking 11 Under Windows XP, the driver ranking 12 strategy has been modified as follows: If an INF file is unsigned, and if neither the 13 [Models] section nor the [DDInstall] 14 section is decorated with an NT-specific extension, the INF file is considered "suspect" and its rank is shifted into a 15 higher range (that is, worse) than all 16 hardware and compatible rank matches of INF files for which one (or both) of those 17 criteria are met. The new ranking ranges will now be: 18 19 0 - 0xFFF(DRIVER HARDWAREID RANK): 20 "trusted" hardware-ID match 0x1000 - 0x3FFF: "trusted" compatible-ID match 21 0x8000 - 0x8FFF: "untrusted" hardware-ID match 22 0x9000 - 0xBFFF: "untrusted" 23 compatible-ID match 0xC000 - 0xCFFF: "untrusted" undecorated hardware-ID match (possibly a 24 Windows 9x-only driver) 0xD000 - 0xFFFF: "untrusted" 25 undecorated compatible-ID match (possibly a Windows 9x-only driver) 26 27 127. A method as in claim 126, in which The authentication information will said authentication information at least in 28 identify Microsoft, operator of the first

apparatus

part identifies said first apparatus and/or a

293482.02

Exhibit E

. 2	2 INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 6,185,683					
3 4 5	126.	Products Infringing: Microsoft Software that includes the Authenticode feature, .NET Framework SDK, Visual Studio, Microsoft technology that supports a digital signature function (such as ActiveX),				
7	A method of providing trusted intermediary services including the following steps:	ActiveX control, Cabinet file, Microsoft				
8		Windows Installer, Authenticode and Software Restriction Policy technologies. For example, a software publisher				
9 10		distributing a signed application that has licensed ActiveX controls embedded within it would practice this method.				
11	at a first apparatus, receiving an item from a second apparatus;	The item is unsigned software such as an ActiveX control or any software packaged				
12		in a cabinet file or Microsoft Installer (.msi) file. Within the development environment, multiple software developers				
13 14		(working on a second apparatus) will send their unsigned software to a secure location (first apparatus) containing the entity's				
15		private signing key. An example entity would be a software publisher.				
16 17		Source: Deploying ActiveX Controls on the Web with the Internet Component Download				
18		The holder of the digital certificate				
19 20		Keeping your digital certificate safe is very important. Some firms (including Microsoft) do not keep their signature file on site. The signature is kept with the				
21		Certificate Authority and files are sent there for signing.				
22 23	associating authentication information with said item;	Signing the software associates the software publisher's identify with the software.				
24 25		Source: Packaging ActiveX Controls Signing Cabinet Files				
26		A .cab file can be digitally signed like an ActiveX control. A digital signature				
27		provides accountability for software developers: The signature associates a software vendor's name with a given file. A				
28		signature is applied to a .cab file (or control) using the Microsoft Authenticode®				
-						

Exhibit B 91

]		technology.
2	2	The .cab tool set assists software developers in applying digital signatures to
. 3		cab files by allowing a developer to
-		allocate space in the .cab file for the
. 4	incorporating said item into a secure digital	signature. Signing software either directly or within a
. 5	11	package (cabinet or .msi file) secures it in a
		digital container.
6	1	Alternately, the signed ActiveX control could be placed into a signed cabinet file.
7	associating a first rule with said secure	The first rule would be the licensing
	digital container, said first rule at least in	support code within the ActiveX control
8	part governing at least one aspect of access to or use of said item;	and/or conditional syntax statements when the software is within a signed .msi file.
9		When the software is within a signed
10		cabinet file, the first rule can be a rule contained in the software, as is the case
10		when an ActiveX control is packaged in a
. 11		signed cabinet file.
12		First rule, in the case of ActiveX:
		3371
13		When an application with a licensed ActiveX control is started, an instance of
14		the control usually needs to be created.
15		The application accomplishes this by making a call to CreateInstanceLic and
13		passing the license key embedded in the
16	·	application as a parameter in the call. The
17		ActiveX control performs a string comparison between the embedded license
		key and its own copy of the license key. If
18		the keys match, an instance of the control is created and the application can execute
19		normally.
-20		Source: Using ActiveX Controls to
ZU	·	Automate Your Web Pages
21	,	Run-time licensing Most Active V Controls should gunner
22		Most ActiveX Controls should support design-time licensing and run-time
		licensing. (The exception is the control that
23		is distributed free of charge.) Design-time licensing ensures that a developer is
24	·	building his or her application or Web page
		with a legally purchased control; run-time
25		licensing ensures that a user is running an application or displaying a Web page that
26		contains a legally purchased control.
	·	Design-time licensing is verified by control containers such as Visual Basic, Microsoft
27		Access, or Microsoft Visual InterDev®.
28	·	Before these containers allow a developer
		to place a control on a form or Web page.

1		
1		they first verify that the control is licensed
2		by the developer or content creator. These
		containers verify that a control is licensed
· 3		by calling certain functions in the control:  If the license is verified, the developer can
٠		add it.
. 4		Run-time licensing is also an issue for
_		these containers (which are sometimes
5		bundled as part of the final application); the
.6		containers again call functions in the
, U		control to validate the license that was
· 7		embedded at design time.
-	transmitting said secure digital container	The third apparatus is a user computer or
. 8	and said first rule to a third apparatus, said	an application server. The protected processing environment (PPE) is Windows
	third apparatus including a protected processing environment at least in part	operating system, Internet Explorer (IE)
9	protecting information stored in said	and pertinent operating IE services such as
10	protected processing environment from	Windows File Protection and security,
10	tampering by a user of said third apparatus;	signature and cryptographic functions
11		related to code signing and related policies.
		The PPE checks for signatures on software or the software packages and detects
12		situations when the signature does not
		validate as an indication that tampering
. 13		may have occurred with the item.
14	said third apparatus receiving said secure	Having the third apparatus receiving said
	digital container and said first rule;	secure digital container and said first rule is
15.		typical of networked computing environments.
	said third apparatus checking said	Examine the signature information includes
16	authentication information; and	verifying that signature was creating using
17	,	the private key that corresponds to the
- '		public key of the publisher.
18	said third apparatus performing at least one	The action would be installation and/or use
	action on said item, said at least one action	of the distributed software. The second rule can be software restriction policies
19	being governed, at least in part, by said first rule and by a second rule resident at	resident on the machine, which can be
20	said third apparatus prior to said receipt of	invoked at installation and/or runtime.
20	said secure digital container and said first	
-21	rule, said action governance occurring at	.NET Framework Security – pg 259
	least in part in said protected processing	
22	environment.	and
		White Paper – Using Software Restriction
23		Policies in Windows XP and Windows
24	·	NET Server to Protect Against
24		Unauthorized Software
25		
		Software Restriction Polices is a policy-
26		driven technology that allows
		administrators to set code-identity-based rules that determine whether an application
27		is allowed to execute. (.NET Framework
20		Security – pg 259)
28		

2	For example, administrators can set rules for all Windows Installer packages coming from the Internet or Intranet zone.
3 4 5	As part of the DLL load mechanisms, Software Restriction Policies is invoked and starts to check its most specific rules. Software Restriction Policies get invoked
6	prior to an .exe being able to run.
7	The four types of rules are – hash, certificate, path, and zone.
8	Note: The hash and certificate rules relate directing to the signature information
9	whereas, the path and zone rules do not.
10	127. A method as in claim 126, in which said authentication information at least in is identified in the authentication
- 11	part identifies said first apparatus and/or a information. user of said first apparatus.
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
- 1	
22	
23	
24	
25	
26	
27	
28	

Product infringing: Visual Studio NET, NET Framework SDK, Authenticode, Products that contain the NET CLR, Compact CLR or CLI.  A method of providing trusted intermediary services including the following steps: at a first apparatus, receiving an item from a second apparatus;  Basociating authentication information with said item;  First apparatus is a software build or deployment services computer that has access to signing key. The item may be a program, graphic, media object or other resource, from a developer computer, or archive (second apparatus).  Associating a cryptographic hash with the file that will contain this item for the purpose of ensuring the authenticity of the item, along with names and attributes that are desired to be associated with the item for identification purposes.  Broducing a graphic media object or other resource, from a developer computer, or archive (second apparatus).  Associating a cryptographic hash with the file that will contain this item for the purpose of ensuring the authenticity of the item, along with names and attributes that are desired to be associated with the item for identification purposes.  Producing signed, strongly named associated attributes.  Including any security demands (such as members of the Microsoft NET Framework SDK Public Class CodeAccessSecurityAttribute) as part of the assembly.  The third apparatus is a user computer or an application server. The third apparatus's protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus; special protected processing environment is windows NT and the NET CLR, CLI and/or compact CLR distributions, and the with the Windows NT and CLR, CLI and/or compact CLR distributions, and secure digital container and said first rule is typical of networked computing environments.  The NET Framework, when the assembly is installed into the global assembly cache (GAC). verifies the strong name of assembly to the proving	. 4		
A method of providing trusted intermediary services including the following steps: at a first apparatus, receiving an item from a second apparatus; receiving an item from a second apparatus; second apparatus; access to signing key. The item may be a program, graphic, media object or other resource, from a developer computer, or archive (second apparatus).  Associating authentication information with said item;  associating authentication information with said item;  associating authentication information with said item;  associating a first rule with said secure digital container;  associating a first rule with said secure digital container, said first rule at least in part governing at least one aspect of access to or use of said item;  transmitting said secure digital container and said first rule to a third apparatus, relating to the assembly. The third apparatus is a user computer or an application server. The third apparatus's protected processing environment from tampering by a user of said third apparatus;  asid third apparatus receiving said secure digital container and said first rule;  asid third apparatus checking said authentication information; and  Compact CLR occll.  First apparatus is a software build or deployment services computer that has access to signing key. The item may be a program, graphic, media object or other resource, from a developer computer, or archive (second apparatus).  Ascociating a developer computer, or archive (second apparatus is a developer computer or an application purposes.  Including any security demands (such as members of the Microsoft .NET Framework SDK Public Class  CodeAccessSecurityAttribute) as part of the assembly.  The third apparatus is a user computer or an application server. The third apparatus's protected processing environment is Windows NT and the .NET CLR, CLI and/or compact CLR, cli and/or compact CLR distributions.  Ametical paparatus receiving said secure digital container and said first rule;  Sociated attributes.  The third apparatus is a user c	•	126.	.NET Framework SDK, Authenticode,
A method of providing trusted intermediary services including the following steps: at a first apparatus; receiving an item from a second apparatus;  Birst apparatus is a software build or deployment services computer that has access to signing key. The item may be a program, graphic, media object or other resource, from a developer computer, or archive (second apparatus).  Associating authentication information with said item;  associating authentication information with said item;  Associating a cryptographic hash with the file that will contain this item for the purpose of ensuring the authenticity of the item, along with names and attributes that are desired to be associated with the item for identification purposes.  Producing signed, strongly named associated attributes.  Including any security demands (such as members of the Microsoft, NET Framework SDK Public Class CodeAccessSecurityAttribute) as part of the assembly.  Transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus; spicated protecting information stored in said protected processing environment from tampering by a user of said third apparatus; spicated from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLT and/or compact CLR distributions.  Having the third apparatus receiving said secure digital container and said first rule is typical of networked computing environments.  The .NET Framework, when the assembly is installed into the global assembly cache (GAC) verifie	. 6		Products that contain the .NET CLR, Compact CLR or CLI.
at a first apparatus, receiving an item from a second apparatus;  10 110 111 112 12		A method of providing trusted intermediary	
a second apparatus;  deployment services computer that has access to signing key. The item may be a program, graphic, media object or other resource, from a developer computer, or archive (second apparatus).  associating authentication information with said item;  associating authentication information with said item;  associating authentication information with said item;  associating a cryptographic hash with the file that will contain this item for the purpose of ensuring the authenticity of the item, along with names and attributes that are desired to be associated with the item for identification purposes.  associating a first rule with said secure digital container, said first rule at least in part governing at least one aspect of access to or use of said item;  transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protected processing environment at least in part protected processing environment at least in part protected processing environment from tampering by a user of said third apparatus;  asid third apparatus receiving said secure digital container and said first rule;  asid third apparatus receiving said secure digital container and said first rule;  said third apparatus checking said authentication information; and  associating a cryptographic hash with the file that will contain this item for the purpose of ensuring the authenticity of the item, along with names and attributes that are desired to be associated with the item for identification purposes.  Including any security demands (such as members of the Microsoft .NET Tensework SDK Public Class CodeAccessSecurityAttribute) as part of the assembly.  The third apparatus is a user computer or an application server. The third apparatus receiving said secure digital container and said first rule; is typical of networked computing environments.  The .NET Framework, when the assembly is installed into the global assembly cache (GAC), ve	,		First apparatus is a software build or
program, graphic, media object or other resource, from a developer computer, or archive (second apparatus).  Associating a cryptographic hash with the file that will contain this item for the purpose of ensuring the authenticity of the item, along with names and attributes that are desired to be associated with the item for identification purposes.  Including any security demands (such as members of the Microsoft .NET part governing at least one aspect of access to or use of said item;  transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;  The third apparatus is a user computer or an application server. The third apparatus's protected processing environment from tampering by a user of said third apparatus;  asaid third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus checking said authentication information; and  program, graphic, from a developer computer, or archive (seconid apparatus).  Associating a cryptographic hash with the file that will contain this item for the purpose of ensuring the authenticity of the item, along with names and attributes that are desired to be associated with the item for identification purposes.  Producing signed, strongly named associated attributes.  Including any security demands (such as members of the Microsoft .NET Framework SDK Public Class CodeAccessSecurityAttribute) as part of the assembly.  The third apparatus is a user computer or an application server. The third apparatus's protected processing environment is Windows NT and the .NET CLR, CLI and/or compact CLR distributions.  Having the third apparatus receiving said secure digit	8		deployment services computer that has
associating authentication information with said item;  12 13 14 15 16 17 18 18 18 19 19 19 10 10 10 10 11 11 11 12 13 14 15 15 16 16 17 17 18 18 18 18 19 19 19 10 10 10 10 11 11 12 12 13 15 16 17 18 18 18 18 18 18 18 18 18 18 18 18 18	. 9		program, graphic, media object or other
said item;  said item;  file that will contain this item for the purpose of ensuring the authenticity of the item, along with names and attributes that are desired to be associated with the item for identification purposes.  producing signed, strongly named assembly that contains this assembly and associated attributes.  associating a first rule with said secure digital container, said first rule at least in part governing at least one aspect of access to or use of said item;  transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus checking said authentication information; and  file that will contain this item for the purpose of ensuring the authenticity of the item, along with names and attributes that are desired to be associated with the item for identification purposes.  Including any security demands (such as members of the Microsoft NET Framework SDK Public Class CodeAccessSecurityAttribute) as part of the assembly.  The third apparatus is a user computer or an application server. The third apparatus's protected processing environment is Windows NT and the .NET CLR, CLI and/or compact CLR. Information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI and/or compact CLR distributions.  All third apparatus receiving said secure digital container and said first rule is typical of networked computing environments.  The .NET Framework, when the assembly is installed into the global assembly cache (GAC) verifies the strong name of assembly exceived assembly cache (GAC) v	10		archive (second apparatus).
purpose of ensuring the authenticity of the item, along with names and attributes that are desired to be associated with the item for identification purposes.  Producing signed, strongly named associated attributes.  associating a first rule with said secure digital container, said first rule at least in part governing at least one aspect of access to or use of said item;  transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment from tampering by a user of said third apparatus;  asaid third apparatus receiving said secure digital container and said first rule;  asaid third apparatus receiving said secure digital container and said first rule;  asaid third apparatus receiving said secure digital container and said first rule;  asaid third apparatus checking said authentication information; and  purpose of ensuring the authenticity of the item, along with names and attributes that are desired to be associated with the item for identification purposes.  Producing signed, strongly named associated attributes.  Including any security demands (such as members of the Microsoft .NET  Framework SDK Public Class  CodeAccessSecurityAttribute) as part of the assembly windows NT and the .NET  CLR, CLI and/or compact CLR.  Information is protected processing environment is windows NT and the .NET  computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI' and/or compact CLR distributions.  Having the third apparatus receiving said secure digital container and said first rule is typical of networked computing environments.  The .NET Framework, when the assembly is installed into the global assembly cache (GAC), verifies the strong name of assembles. This process includes verifying that signature was creating using			Associating a cryptographic hash with the file that will contain this item for the
incorporating said item into a secure digital container;  Producing signed, strongly named assembly that contains this assembly and associated attributes.  Is associating a first rule with said secure digital container, said first rule at least in part governing at least one aspect of access to or use of said item;  transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;  Code Access Security Attribute) as part of the assembly.  The third apparatus is a user computer or an application server. The third apparatus's protected processing environment is Windows NT and the .NET CLR, CLI and/or compact CLR.  Information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI and/or compact CLR distributions.  Having the third apparatus receiving said secure digital container and said first rule is typical of networked computing environments.  The .NET Framework when the assembly is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using	••		purpose of ensuring the authenticity of the
incorporating said item into a secure digital container; said first rule at least in part governing at least one aspect of access to or use of said item;  transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected protecting information stored in said protected processing environment from tampering by a user of said third apparatus; said third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule is typical of networked computing environments.  The NET Framework, when the assembly is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using	12		item, along with names and attributes that
incorporating said item into a secure digital container;  associating a first rule with said secure digital container, said first rule at least in part governing at least one aspect of access to or use of said item;  transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protected processing environment from tampering by a user of said third apparatus;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus checking said authentication information: and  Producing signed, strongly named assembly that contains this assembly assembly that contains this assembly and assembly that contains this assembly assembly that contains this assembly assembly that contains this assembly assembly that contains this assembly assembly that contains this assembly assembly that contains this assembly and assembly that contains this assembly assembly that contains this assembly assembly that contains this assembly assembly that contains this assembly assembly that contains this assembly assembly that contains this assembly assembly that contains this assembly associated attributes.  Including apy security demands (such as members of the Microsoft .NET Framework SDK Public Class CodeAccessSecurityAttribute) as part of the assembly and associated attributes.  The third apparatus is a user computer or an application server. The third apparatus's protected processing environment is Windows NT and the .NET CLR, CLI and/or compact CLR distributions.  Having the third apparatus receiving said secure digital container and said first rule is typical of netw			
assembly that contains this assembly and associating a first rule with said secure digital container, said first rule at least in part governing at least one aspect of access to or use of said item;  Transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;  20	13	incorporating said item into a secure digital	
associating a first rule with said secure digital container, said first rule at least in part governing at least one aspect of access to or use of said item;  transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;  Clap color of the assembly.  The third apparatus is a user computer or an application server. The third apparatus's protected processing environment is Windows NT and the .NET CLR, CLI and/or compact CLR. Information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI and/or compact CLR distributions.  Additional container and said first rule;  said third apparatus receiving said secure digital container and said first rule is typical of networked computing environments.  The shird apparatus is a user computer or an application server. The third apparatus's protected processing environment is Windows NT and the .NET CLR, CLI and/or compact CLR.  Information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI and/or compact CLR.  Having the Hird apparatus receiving said secure digital container and said first rule is typical of networked computing environments.  The .NET Framework, when the assembly is installed into the global assembly cache (GAC). verifies the strong name of assemblies. This process includes verifying that signature was creating using	14		assembly that contains this assembly and
part governing at least one aspect of access to or use of said item;  transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;  20  21  22  23  24  25  26  27  28  28  29  20  20  20  21  22  23  24  25  26  26  27  28  28  28  29  20  20  20  20  21  22  23  24  25  26  26  27  28  28  28  29  20  20  20  20  21  22  23  24  25  26  26  27  28  28  28  29  20  20  20  20  20  21  22  23  24  25  26  27  28  28  28  29  29  20  20  20  20  20  20  20  20	15	associating a first rule with said secure	Including any security demands (such as
to or use of said item;  CodeAccessSecurityAttribute) as part of the assembly.  Transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;  CodeAccessSecurityAttribute) as part of the assembly.  The third apparatus is a user computer or an application server. The third apparatus's protected processing environment is Windows NT and the .NET CLR, CLI and/or compact CLR. Information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI' and/or compact CLR distributions.  Having the third apparatus receiving said secure digital container and said first rule is typical of networked computing environments.  The .NET Framework, when the assembly is installed into the global assembly cache (GAC). verifies the strong name of assemblies. This process includes verifying that signature was creating using	16	part governing at least one aspect of access	
transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;  20			
and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;  21	17		the assembly.
third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;  21 22 23 24 said third apparatus receiving said secure digital container and said first rule;  25 said third apparatus checking said authentication information; and  26 said third apparatus checking said authentication information; and  27 28 29 third apparatus including a protected processing environment is Windows NT and the .NET CLR, CLI and/or compact CLR. Information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI and/or compact CLR distributions.  26 Having the third apparatus receiving said secure digital container and said first rule is typical of networked computing environments.  27 The .NET Framework, when the assembly is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using			The third apparatus is a user computer or
processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;  21 22 23 24 said third apparatus receiving said secure digital container and said first rule;  25 said third apparatus checking said authentication information; and  26 said third apparatus checking said authentication information; and  27 28 29 20 environment is Windows NT and the .NET CLR, CLI and/or compact CLR. Information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI and/or compact CLR distributions. Further this information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected from tampering because user is not administrator, user runs code on server, a share on another computer.  24  25  26  27  28  28  28  29  20  21  22  23  24  25  26  27  28  28  29  20  20  21  21  22  23  24  25  26  27  28  28  29  20  20  21  21  22  23  24  25  26  27  28  28  29  20  20  21  21  22  23  24  25  26  27  28  29  20  20  21  21  22  23  24  25  26  27  28  29  20  20  21  21  22  23  24  25  26  27  28  29  20  20  20  20  20  21  21  22  23  24  25  26  27  27  28  29  20  20  20  20  20  20  21  21  22  23  24  25  26  27  27  28  29  20  20  20	18		an application server. The third
protecting information stored in said protected processing environment from tampering by a user of said third apparatus;  21 22   23   24   said third apparatus receiving said secure digital container and said first rule;  25   26   said third apparatus checking said authentication information; and  26   27   28   28   28   29   20   20   20   20   20   20   20			
protected processing environment from tampering by a user of said third apparatus;  Information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI and/or compact CLR distributions.  Having the third apparatus receiving said secure digital container and said first rule;  said third apparatus checking said authentication information; and  The .NET Framework, when the assembly is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using	19		
tampering by a user of said third apparatus;  because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI and/or compact CLR distributions.  Having the third apparatus receiving said secure digital container and said first rule;  said third apparatus checking said secure digital container and said first rule is typical of networked computing environments.  The NET Framework, when the assembly is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using	20.		
code on server, a share on another computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI and/or compact CLR distributions.  Said third apparatus receiving said secure digital container and said first rule;  Having the third apparatus receiving said secure digital container and said first rule is typical of networked computing environments.  Said third apparatus checking said authentication information: and  The .NET Framework, when the assembly is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using	.20	tampering by a user of said third apparatus:	
computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI and/or compact CLR distributions.  24 said third apparatus receiving said secure digital container and said first rule;  25 Having the third apparatus receiving said secure digital container and said first rule is typical of networked computing environments.  26 said third apparatus checking said authentication information; and  27 The .NET Framework, when the assembly is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using	21	1 8 3	
information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI and/or compact CLR distributions.  24 said third apparatus receiving said secure digital container and said first rule;  25 Having the third apparatus receiving said secure digital container and said first rule is typical of networked computing environments.  26 said third apparatus checking said authentication information; and  27 The .NET Framework, when the assembly is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using			
with the Windows NT and CLR, CLI' and/or compact CLR distributions.  Having the third apparatus receiving said secure digital container and said first rule;  said third apparatus checking said secure digital container and said first rule is typical of networked computing environments.  The .NET Framework, when the assembly is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using	22		information is protected by a number of
said third apparatus receiving said secure digital container and said first rule;  Having the third apparatus receiving said secure digital container and said first rule is typical of networked computing environments.  Said third apparatus checking said authentication information; and  The .NET Framework, when the assembly is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using			protection mechanisms that are included
24 said third apparatus receiving said secure digital container and said first rule; 25 said third apparatus checking said secure digital container and said first rule is typical of networked computing environments.  26 said third apparatus checking said authentication information; and is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using	23		
digital container and said first rule;  secure digital container and said first rule is typical of networked computing environments.  Said third apparatus checking said authentication information; and is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using	24	said third apparatus receiving said secure	Having the third apparatus receiving said
26 said third apparatus checking said authentication information; and 27 (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using	j	digital container and said first rule;	secure digital container and said first rule is
said third apparatus checking said authentication information; and is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using	25		
authentication information; and is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using	26	said third apparatus checking said	
27 (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using			is installed into the global assembly cache
assemblies. This process includes verifying that signature was creating using	27		(GAC), verifies the strong name of
verifying that signature was creating using			assemblies. This process includes
the private key that corresponds to the	28	<u> </u>	verifying that signature was creating using
	ļ.	<u></u>	the private key that corresponds to the

Exhibit B

1

2

	•	• •			. ,		,		٠.		•
said third apparatus performing at least one action on said item, said at least one action being governed, at least in part, by said first rule and by a second rule resident at said third apparatus prior to said receipt of said secure digital container and said first rule, said action governance occurring at least in part in said protected processing environment.				at of rst at	public key of the publisher.  The action is executing code that is the item or using code that renders the item. Action is governed by security demands on code that calls the item or on code that calls code included in the .NET assembly that manages said item. The second rule is the machine, enterprise, user, and application configuration file resident rules. Typically these configuration files will be populated before the arrival of most new assemblies in a virtual distribution environment. This action governance occurs in the protected processing environment of the CLR, CLI and/or compact CLR.						
said part	authentic identifies	od as in cl cation info s said first rst appara	ormation tapparat	at least i	in	The auther identify the company is that identifies apparentity or a	e .NET name and the first the and the	Assend tradespead	nbly Clemark us or us a meml	lass attrib ser of ber of	the an
	•	-					•				
				•	•						•
٠					•						
		• .									
					•				-		
				•							
					, .		•				
				•				-			• .
	•	•									
	•			•					•		
				•	·	•	•				
	,			•	٠			-			: • `
	٠.									•	
	•										
	·										
				,							•
			_								

_	•	T .
4 5 6	126.	Product infringing: Visual Studio .NET, .NET Framework SDK, Authenticode, Products that contain the .NET CLR, Compact CLR or CLI.
	A method of providing trusted intermediary services including the following steps:	
7 8 9	at a first apparatus, receiving an item from a second apparatus;	The item is an unsigned .NET assembly, which can include, but not be limited to, a Web control, multi-file assembly or component. Within the development environment, multiple assembly builders
10 11		(working on a second apparatus) will send their unsigned assembly to a secure location (first apparatus) containing the entity's private signing key. An example
12		entity would be a software publisher.
- 13		.NET Security Framework - pg 130-1
14		Describes this exact practice and further explains the "Delay Signing Assemblies" feature of .NET that accommodates the fact
15 16		that "many publishers will keep the private key in a secure location, possibly embedded in specially designed
17.		cryptographic hardware."
18		"Delay signing is a technique used by developers whereby the public key is added
19		to the assembly name as before, granting the assembly its unique identity, but no
*20		signature is computed. Thus, no private key access is necessary."
21	associating authentication information with said item;	Strong naming the assembly binds the entity's/publisher's name into the
22		assembly. The public portion of the key used to strongly name the assembly is
23		placed in the assembly manifest. Other assemblies or applications can contain
24		references to the strong names of strongly named assemblies such as in the case of
25		applications that contain references to a set of compliant .NET core libraries. Strong
26		naming compliant .NET core libraries with the European Computers Manufactures
27	,	Association's (ECMA) key is a way to allow any publisher to develop compliant
28		NET core libraries that can be authenticated by other applications.
· 1		ii

Exhibit B

293482.02

2

	. ∬	
2	. 🛮	NET Security Framework - pg 124
	.	"Strong naming is a process whereby an
3	•	assembly name can be further qualified by the identity of the publisher."
	1	NET Security Framework – pg 133
4		The publisher must advertise its public key
5		or keys in an out-of-band fashion (such as
3		documentation shipped with the product or
. 6		on the company Web site)
. 0		.NET Security Framework - pg 130
7		The goal of the ECMA key is to allow a
<b>.</b> .		slightly more generalized strong name
8		binding than usual, namely allowing
		binding to the publisher of the runtime in
9		use, rather than to a fixed publisher.
	incorporating said item into a secure digital	Signing the assembly places it in a secure
10	container;	container.  NET Framework Security – pg 527
	- 1	Strong named assemblies cannot be
11		modified in any manner without destroying
12		the strong name signature.
12	· ·	Applied Microsoft .NET Framework
13		Programming – pg 89
•		Strongly Named Assemblies Are Tamper-
14		Resistant
	1	When the assembly is installed into the
15		GAC, the system hashes the contents of the
		file containing the manifest and compares the hash value with the RSA digital
16		signature value embedded within the PE
17		file (after unsigning it with the public key).
17		If the values are identical, the file's
18		contents haven't been tampered with and
		you know that you have the public key that
19		corresponds to the publisher's private key.
		In addition, the system hashes the contents
20		of the assembly's other files and compares the hash values with the hash values stored
		in the manifest file's FileDef table. If any
-21		of the hash values don't match, at least one
22		of the assembly's files has been tampered
22		with and the assembly will fail to install
.23		into the GAC.
.23	associating a first rule with said secure	A .NET assembly includes imperative and
24	digital container, said first rule at least in	declarative statements/rules that will
	part governing at least one aspect of access	govern its access or use. For example,
25	to or use of said item;	role-based security or strong name
		demands in the assembly can be the first
26		rule.
]		MSDN on Pole Posed Security
27	 	MSDN on Role-Based Security
_		Applications that implement role-based
28		security grant rights based on the role
ı		ii
		<b>.</b>

l		associated with a principal object. The
_		principal object represents the security
- 2		context under which code is running. The
3		PrincipalPermission object represents the
		identity and role that a particular principal.
· 4		class must have to run. To implement the
. •		PrincipalPermission class imperatively,
-		create a new instance of the class and
· 5		initialize it with the name and role that you
	·	want users to have to access your code.
6		want about to have to access your code.
		MCDN on StrongNameIdentityParmission
· 7	1	MSDN on StrongNameIdentityPermission
		0. 37 71 7
- 8		StrongNameIdentityPermission class
_		defines the identity permission for strong
9.		names. StrongNameIdentityPermission
,	4	uses this class to confirm that calling code
10		is in a particular strong-named assembly.
10		
• •	transmitting said secure digital container	The third apparatus is a user computer or
11	and said first rule to a third apparatus, said	an application server. The software
12	third apparatus including a protected	publisher transmitting the .NET assembly
	processing environment at least in part	to an end-user with a CLR. The third
13	protecting information stored in said	apparatus's protected processing
	protected processing environment from	environment is Windows NT and the .NET
14	tampering by a user of said third apparatus;	CLR, CLI and/or compact CLR.
- '	·	Information is protected from tampering
15		because user is not administrator, user runs
13		code on server, a share on another
16		computer, or over a network. Further this
10		information is protected by a number of
10		protection mechanisms that are included
17		with the Windows NT and CLR, CLI
		and/or compact CLR distributions.
18	said third apparatus receiving said secure	The end-user receiving the signed
19	digital container and said first rule;	assembly.
	said third apparatus checking said	The NET Framework, when the assembly
20	authentication information; and	is installed into the global assembly cash
-		(GAC), verifies the strong name of
21	,	assemblies. This process includes
		verifying that signature was creating using
22		the private key that corresponds to the
44		public key of the publisher.
		Applied Microsoft .NET Framework
23		Programming ng 90
1		Programming – pg 89
24		Strongly Named Assemblies Are Tamper-
ļ	·	Resistant
25	,	As above.
		•
26		.NET Framework Security – pg 128
20	· i	
<u>,                                    </u>		The verification of any strong name
27		assemblies is performed automatically
i		
28		when needed by the .NET Framework.
		Any assembly claiming a strong name but
l		

	1		
	2		failing verification will fail to install into the global assembly or download cache or
	3	said third apparatus performing at least one	Within the CLR (protected processing
	4	action on said item, said at least one action being governed, at least in part, by said	environment), the execution of the program will depend upon whether the user is of the
•	5	first rule and by a second rule resident at said third apparatus prior to said receipt of	fole required of the assembly or whether
-	· #	said secure digital container and said first	the calling assembly is from a strong- named assembly specified in the "item"
	- 1	rule, said action governance occurring at least in part in said protected processing	assembly (alternate first rules) and only if assembly complies with the local code
	7	environment.	access security policy (second rule), as an example of one of the types of rules that
	8		.NET Framework allows to be resident on the third apparatus
:	9.	127. A method as in claim 126, in which	
10	0 113	said authentication information at least in part identifies said first apparatus and/or a	The user of the first apparatus is the developer at the assembly developer. Strong naming
11	ı   <u>[</u>	user of said first apparatus.	binds the publisher's name to assembly.
12		LaMacchia, Brian, etc, <u>NET Framework Secu</u>	urity, Addison-Wesley, 2002
13	11 4	Cichter, Jeffrey, Applied Microsoft NET Fran	nework Programming, Microsoft Press, 2002
14			
15			· ·
16			
17			
18			
19			
20			
21			
22 23			
23			
24			
25			•
26			
27			
i			
28			

	3	IENI NO. 6,253,193
•	CLAIMEANGUAGE AND AND AND AND AND AND AND AND AND AND	SEE CLAIM OF THE RINGEMENT SERVE
•	5	Infringing products include Windows Media Player and Windows Media Rights Manager SDK
	A method comprising:	SDR
	6 (a) receiving a digital file including music;	Reference is made to the Windows Media
	7	("WMRM SDK"), attached hereto as Exhibit
,	8	forth herein using the example of a music file
4	9	player.
- 10	(h) storing soid dicital St.:	Consumer receives a Windows Media file (WMRM SDK, Step 3)
11	(b) storing said digital file in a first secure memory of a first device;	Windows Media file is stored in consumer's
12		computer and all use of it is securely managed by the Secure Content Manager in Windows Media Player.
13	(c) storing information associated with said	License is stored in the License Store (WA(D)
	digital file in a secure database stored on said first device, said information including at least	1 De la dich die license inclinées Righte which
14	Il one budget control and at least one conv	I may include Allow I tangterto Nonginkar
	control, said at least one budget control	AllowTransfertoSDMI, (or Allow Transfer to
15	I including a budget specifying the number of	WM-D-DRM-Compliant devices or other types of devices), and TransferCount- the
16	Copies which can be made of said digital file.	I import of times a piece of content may be
10	and said at least one copy control controlling the copies made of said digital file;	transferred to the device (a transfer budget).
17	(d) determining whether said digital file may	
18	on at least said copy control.	Windows Media Rights Manager enforces the license restrictions
19	(e) if said copy control allows at least a portion of said digital file to be copied and stored on a	Windows Media Rights Manager determines
	second device,	Whether the Allow I ransfer To Non Chian
20		Allow Transfer to WM D. D. C. C.
21		Allow Transfer to WM-D-DRM-Compliant devices or other types of devices.)
21	(1)copying at least a portion of said digital	Transfer to the SDMI or non-SDMI portable
22	file;	device (Allow I ransfer to WM_D_DRM_
		Compliant devices or other types of devices : c
23	(2)transferring at least a portion of said	Landwed by Windows Media Rights Manager II
, 1	digital file to a second device.	Portable device necessarily includes at least a memory and audio output
24	including a memory and an audio	ory and audio output
25	and/or video output:	
	(3)storing said digital file in said memory	Music file is transferred to the portable device
26	of said second device; and (4)including playing said	
~~	(4)including playing said music through said audio output.	Portable device plays the music
	2. A method as in claim 1, further	
28	comprising: (a) at a time substantially contemporaneous	
	with said transferring step, recording in said	Counter reflecting TransferCount is
	end authorized and authorized aut	decremented by Windows Media Rights
	Exhib	it B
11	10	1 2

293482.02

lf.	•		•			
	·)			)		
first device info	rmation indicati	ing that said	Manager			
transfer has occu	urred.					
3. A method as	in claim 2, in v	which:				
(a) said information	tion indicating t	hat said	Counter dec	rement redu	ices the all	owable
transfer has occu	irred includes a	n encumbrance	number of b			
on said budget.			•			
4. A method as	<u>in claim 3, in v</u>	vhich:				<del></del>
(a) said encumbi	rance operates to	o, reduce the	Counter deci	rement redu	ces the all	owable
number of copie	s of said digital	file authorized	number of b			
by said budget.		•		<u> </u>		
		•	•			
					•	
	•					
	•	•	•			•
				• .		
•			٠,	•	•	
*				•		
	•		•	ζ.	•	
•		:	•			
			••			
•						•
	•	• •				
					•	

<u> </u>	CNT NO. 6,253,193
	Infringing products include Windows Medi Player and Windows Media Rights Manage SDK
11. A method comprising:	• • •
(a) receiving a digital file;	Consumer receives a Windows Media file (WMRM SDK, Step 3)
(b) storing said digital file in a first secure memory of a first device;	Windows Media file is stored in consumer's computer and all use of it is securely manage by the Secure Content Manager in Windows Media Player.
(c) storing information associated with said digital file in a secure database stored on said	License information is stored in the License Store (WMRM SDK, Step 10), license
first device, said information including a first control;	information includes Rights. License Rights may include AllowTransferToNonSDMI,
	AllowTransferToSDMI (Allow Transfer to WM-D-DRM-Compliant devices or other types of devices), TransferCount
(d) determining whether said digital file may be copied and stored on a second device based	WMRM determines whether transfer rights included in license (WMRM SDK, Step 5)
on said first control,	
(1) said determining step including identifying said second device and determining whether said first control	Portable Device Service Provider Module identifies the portable device as either SDM compliant or non-SDMI-compliant (or WM
allows transfer of said copied file to said second device, said determination	DRM Compliant or other types of supported devices) and provides this information to
based at least in part on the features present at the device to which said	Windows Media Device Manager, which allows the transfer based on whether the dev
copied file is to be transferred; (e) if said first control allows at least a portion	identification matches the License Right.  If Windows Media Rights Manager determine
of said digital file to be copied and stored on a second device,	whether the AllowTransferToNonSDMI or AllowTransferToSDMI rights are present (o
	Allow Transfer to WM-D-DRM-Compliant devices or other types of devices), the
	following steps are performed:
(1) copying at least a portion of said digital file;	Transfer to the SDMI or non-SDMI (Allow Transfer to WM-D-DRM-Compliant or othe portable device, if allowed by Windows Me
	Rights Manager
(2) transferring at least a portion of said digital file to a second device including a memory and an audio	Portable device necessarily includes at least memory and audio output
and/or video output; (3) storing said digital file in said memory of said second device; and	Music file is stored in the portable device
(4) rendering said digital file through said output.	Portable device plays the music

2	<u>INTERTRUST TECHNOLOGI</u>	ES CORP. v. MICROSOFT CORP.
. 3	INTERTRUST INFI	RINGEMENT CHART
,	FOR U.S. PATE	ENT NO. 6,253,193
. 4		
•		Product infringing: Windows Media Player,
5		Windows Media Player, Windows Media
		Rights Manager SDK
. 6	15. A method comprising:	
	(a) receiving a digital file;	Consumer receives a Windows Media file
7		((WMRM SDK, Step 3)
	(b) an authentication step comprising:	<del> </del>
. 8	(1) accessing at least one identifier	License includes identity of user's Windows
	associated with a first device or with a	Media Player. WM Players capable of playing
9	user of said first device; and	protected content must be individualized.
		They contain a unique (Individualized) DRM
10		client component to which protected WMA content licenses are bound. Content licenses
11		are bound to this DRM individualization
11	·	module as the result of a challenge sent from
12		the Client to the WMLM service. The
14	·	challenge contains information about
13		Individualized DRM Client (in the form of an
••		encrypted Client ID) and capabilities of the
14		machine (e.g. support for Secure Audio Path
		(SAP), version of the WMRM SDK supported
15		in the player.
	(2) determining whether said identifier is	Music file cannot be used unless identifier
16	associated with a device and/or user	indicated in License matches user's Windows
10	authorized to store said digital file;	Media Player identifier (that is, the Individualized DRM Client to which the
17	·	license is bound must be the same one
18	·	supported by the device).
10	(c) storing said digital file in a first secure	Music file will not be processed through
19	memory of said first device, but only if said	Windows Media Player, including protected
17	device and/or user is so authorized, but not	rendering buffers, unless the identifiers match.
20	proceeding with said storing if said device	Protected WMA file can be stored on client
	and/or user is not authorized;	even if unauthorized but it cannot be decrypted
.21	•	and enter into the secure boundary (first secure
		memory) of the player unless appropriately
22		licensed.
	(d) storing information associated with said	License includes Rights and is stored in the
23	digital file in a secure database stored on said	License Store, Rights may include
_	first device, said information including at least	AllowTransferToNonSDMI,
24	one control;	AllowTransferToSDMI, (or Allow Transfer To
ا ہے		WM-D-DRM-CompliantDevice or other
25	(a) data ining and advanced district C1.	device) TransferCount
~	(e) determining whether said digital file may	Windows Media Rights Manager enforces the
26	be copied and stored on a second device based	license restrictions
22	on said at least one control;	If appropriate rights are present the fall-wine
27	(f) if said at least one control allows at least a	If appropriate rights are present, the following
20	portion of said digital file to be copied and	steps are performed:
28	stored on a second device.  (1) copying at least a portion of said	Transfer to the SDMI or non-SDMI (or WM-
.	(1) copying at icast a portion of said	L TIMESTER OF THE STAND OF HOR STAND OF MINI-

293482.02

. 1		
2	digital file;	D-DRM Compliant or other) portable device, if allowed by Windows Media Rights Manager
3	(2) transferring at least a portion of said digital file to a second device	Portable device necessarily includes at least a memory and audio output
4	and divinity and an audio	
5	(3) storing said digital file in said memory of said second device; and	Music file is stored in the portable device
6	(4) rendering said digital file through said output.	Portable device plays the music
7	16. A method as in claim 15, in which: said digital file is received in an encrypted	Destorted Windows Madia Pil.
•	form;	Protected Windows Media File is encrypted. WMP will not decrypt file until license is
8	and further comprising:	processed. Licenses are bound to Individualization DLLs, which are bound to
.9	decrypting said digital file after said	Hardware ID. Ind. DLL and Hardware ID must be verified as the lds to which the license
10	authentication step and before said step of storing said digital file in said memory of said	is bound – this is the authentication process.  (Recall that this module was created based in
11	first device.	part on receipt of the Client Hardware ID or
12		fingerprint and the license was create based in part on receipt of a challenge from the client
13		indicating the security properties (SAP-ready, SDK support, etc.) of the client).
14		
15		
-16		
17		•
18	:	
19		
20		
21		
i	·	
22		
23		
24	-	
25		
26		
- 11		<u>,</u>

293482.02

27

	FOR 0.3. 1A11	ENT NO. 0,233,173
3	CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
5 6	19.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
	A method comprising:	
7 8 9	receiving a digital file at a first device;	Receiving a digital file such as a Word Document, email, Excel spreadsheet, PowerPoint presentation, or other content at a recipient's device. Such content may be
-		received via email, received on removable media, such as floppy disk, downloaded and
10		viewable by Internet Explorer, e.g., a web page possibly containing graphics and/or audio data,
	ostablishing communication between said first	lf the digital file is subject to rights
12 13	establishing communication between said first device and a clearinghouse located at a location remote from said first device;	management, and the recipient tries to open the digital file in an IRM-enabled application, the
٠.		IRM-enabled application contacts a remote RMS, i.e., clearinghouse for a use license.
. 14	said first device obtaining authorization	If the recipient is authorized to access or use
15	information including a key from said clearinghouse;	the digital file, the RMS creates a license for the digital file. The RMS then seals a key
16		inside the license so that only the recipient canaccess or use the digital file. Finally, the RMS sends the license back to the recipient.
17	said first device using said authorization	The recipient's device then uses the key in the
18	information to gain access to or make at least one use of said first digital file, including	license to gain access or decrypt a portion of the digital file.
19	using said key to decrypt at least a portion of said first digital file; and	
20	receiving a first control from said	The license received from the RMS at the
21	clearinghouse at said first device;	recipient's device contains at least one control, such as restricting the ability to print, forward, or edit.
22	storing said first digital file in a memory of said first device;	The digital file is stored in the memory of the said recipient's device, such as in RAM, on a hard drive, etc.
رے	using said first control to determine whether	The at least one control in the license limits
24	said first digital file may be copied and stored on a second device;	copying the digital file.
25	•	Such controls are set when the digital file was authored. For example, when the digital file is
26		authored, the IRM-enabled application presented the author with a list of policy
27		templates with different rights levels. The
28		author selected an appropriate rights level which may for instance, allow other users in the system to open and read the document, but not
	** II	
3.		12

Exhibit E

1		
2		to modify it, copy text from it, or forward it. These rights or controls are then associated with the digital file.
3 4 5 6		When an attempt is made to access the digital file, the RMS determines the recipient's rights based on the recipient's identity and the policies or controls associated with the digital file.
7	if said first control allows at least a portion of said first digital file to be copied and stored on	If the control in the license allows copying the digital file to a second device, then at least a
. 8	a second device, copying at least a portion of said first digital	portion of the digital file is copied, such as by transferring or forwarding the digital
9	file;	file in an email message;
• 9	transferring at least a portion of said first digital file to a second device including a	A portion of the digital file is then transferred to a second device, such as a personal computer
10	memory and an audio and/or video output;	or portable device. The second device includes
11		a memory and an audio and/or video output. The memory may be a hard-drive, RAM, CD,
12		DVD, or other storage. The audio and/or video
13	storing said first digital file portion in said	output may be speakers and/or a video monitor.  The digital file is stored in the second device's
14	memory of said second device; and rendering said first digital file portion through	memory.
15	said output.	The digital file is rendered through the output, such as played through the speakers and/or displayed on the video monitor. For example, a
16		Word document is displayed on the screen of the video monitor.
		Word document is displayed on the screen of
. 17		Word document is displayed on the screen of
17 18		Word document is displayed on the screen of
. 17		Word document is displayed on the screen of
17 18		Word document is displayed on the screen of
17 18 19		Word document is displayed on the screen of
17 18 19 20		Word document is displayed on the screen of
17 18 19 20 21 22		Word document is displayed on the screen of
17 18 19 20 21 22 23		Word document is displayed on the screen of
17 18 19 20 21 22 23 24		Word document is displayed on the screen of
17 18 19 20 21 22 23 24 25		Word document is displayed on the screen of
17 18 19 20 21 22 23 24		Word document is displayed on the screen of
17 18 19 20 21 22 23 24 25		Word document is displayed on the screen of
17 18 19 20 21 22 23 24 25 26		Word document is displayed on the screen of
17 18 19 20 21 22 23 24 25 26 27		Word document is displayed on the screen of

293482.02

# INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP. INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 6,253,193

	-		
<b>4</b> 5	:		Infringing products include Windows Media Player, Windows Media Rights Manager SDK
,	19	. A method comprising:	•
6	(a)	receiving a digital file at a first device;	WMRM SDK, Step 3.
U	(b)	establishing communication between said	WMRM SDK, Step 6.
7	<b>\</b> \`	first device and a clearinghouse located at	
,	H	a location remote from said first device;	
8	(c)	said first device obtaining authorization information including a key from said clearinghouse;	WMRM SDK, Step 9. [License contains the key]
9	(4)		WMRM SDK, Step 11.
	(d)	information to gain access to or make at	WINION SDR, Step 11.
.10	1	least one use of said first digital file,	
	1	including using said key to decrypt at least	
11	1	a portion of said first digital file; and	
12	(e)	receiving a first control from said	WMRM SDK, Steps 8-9.
12		clearinghouse at said first device;	
13	(f)	storing said first digital file in a memory of said first device;	WMRM SDK, Step 3.
14	(g)	using said first control to determine	At least the following WMRMRights Object
		whether said first digital file may be	properties meet this limitation:
15		copied and stored on a second device;	AllowTransferToNonSDMI,
			AllowTransferToSDMI (or AllowTransfer To
16			WM-D-DRM-Compliant Device or other) and
	<u> </u>		TransferCount TransferCount
17	(h)	if said first control allows at least a portion	This and all subsequent claim steps occur when
		of said first digital file to be copied and	the condition specified in the WMRMRights
18	(3)	stored on a second device,	Object property is met Transfer to the SDMI or non-SDMI (or WM-
19	(i)	copying at least a portion of said first digital file;	D-DRM Compliant) portable device, if
ן עו		uigitai tiie,	allowed by Windows Media Rights Manager
20	(i)	transferring at least a portion of said first	Portable device necessarily includes at least a
20	W	digital file to a second device including a	memory and audio output
21		memory and an audio and/or video output;	momory and address output
~ 1	(k)	storing said first digital file portion in said	Music file is stored in the portable device
22	(4)	memory of said second device; and	Tradit in the board in the periods
	(1)	rendering said first digital file portion	Portable device plays the music
23	(i)	through said output.	A straight and prayer and arrange
		and other contractions	
24			· .
ŀ	1		· 1

# INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP. INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 6,253,193

4		Infringing products include Windows Media Player, Windows Media Player, Windows Media Rights Manager SDK
5	51. A method comprising:	Wedia Rights Manager SDR
6	(a) receiving a digital file at a first device;	WMRM SDK, Step 3.
7	(b) establishing communication between said first device and a	WMRM SDK, Step 6.
8	clearinghouse located at a location remote from said first device;	
9	(c) said first device obtaining authorization information from said	WMRM SDK, Step 9.
10	clearinghouse; and (d) said first device using said	WMRM SDK, Step 11.
11	authorization information to gain access to or make at least one use of said first digital file:	
12	(e) storing said first digital file in a	WMA file stored on client
13	memory of said first device; (f) using at least a first control to	If device is based on WM D-DRM, it has a
14	determine whether said first digital file may be copied and stored on a second	certificate that is used to identify the device as compliant as well as the device's security
15	device, said determination based at least in part on (1) identification information	level. The security level indicates support on the device for such attributes as an internal
16.	regarding said second device, and (2) the functional attributes of said second	clock.
17	device:	If License specifies that transfer of protected
18	(g) if, based at least in part on said identification information, said first control allows at least a portion of said	WMA file to WM-D-DRM-Compliant device is allowed, transfer may occur.
19	first digital file to be copied and stored on a second device,	io anomou, dansior may coolar
20	(h) copying at least a portion of said first digital file;	If transfer is a licensed right as indicated in the license, the song is copied to the device via
21		Windows Media Device Manager.
22	(i) transferring at least a portion of said first digital file to a second device	Windows Media Device Manager transfers the content to the device:
23	including a memory and an audio and/or video output;	
24	(j) storing said first digital file portion in said memory of said second device;	WMA file is stored on device
25	and (k) rendering said first digital file	WMA file is rendered.
26	portion through said output.	

27

2

28

্হ

## INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP. INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 5,915,019

3.			
	MANUACCE AM LANGUAGE MANUACCE AND ACCESS OF THE PARTY OF	WWW. CEATM OF INFRINGEMENTS	Jis.
<ul><li>4</li><li>5</li><li>6</li></ul>	33.	Infringing products include all Microsoft tools that support the Microsoft ActiveX licensing model, Visual Studio .NET, the Microsoft Installer SDK, and Operating System products that include the Microsoft Installer technology.	
7	A data processing arrangement comprising at least one storing arrangement that at	The first protected data is an ActiveX control.	
8 9	least temporarily stores a first secure container comprising first protected data and a first set of rules governing use of said	The first alternative for the first secure container is the signed .msi in which the	
10	first protected data,	ActiveX developer packaged the ActiveX control. The first set of rules is the	
11		conditional syntax statements of the signed .msi file.	·
12		The second alternative for the first secure container is the signed and licensed	
13		ActiveX control. The first set of rules is the license support code in the ActiveX	
14		A third alternative for the first container is	
15 16		a signed cabinet file containing a (signed or unsigned) ActiveX control with license	
17		support code. The first set of rules is the license support code in the ActiveX control.	
18	·	condo.	
19	and at least temporarily stores a second	The second protected data is the application	
20	secure container comprising second protected data different from said first protected data and a second set of rules	developer's application that includes/uses the ActiveX control. The application developer's signed .msi file (second secure	
21	governing use of said second protected data; and	container) contains the application (second protected data). The second set of rules is	
22		the signed .msi file's conditional syntax statements that will be governed the offer/installation of the application.	
23	a data transfer arrangement, coupled to at	Placing the licensed ActiveX control (first	
24	least one storing arrangement, for transferring at least a portion of said first	protected information) in a signed cabinet file (third secure container) that itself is	
25	protected data and a third set of rules governing use of said portion of said first	included in the application's signed .msi file (second secure container). The third	
26	protected data to said second secure container.	set of rules is the license support code in the ActiveX control.	
27	further comprising		
28	means for creating and storing, in said at least one storing arrangement, a third secure container;	The ability of the application developer to package files in signed cabinet files.	

Exhibit B

293482.02

said data transfer arrangement further The third secure container is a cabinet file comprising means for transferring said signed by the application developer and 2 portion of said first protected data and including at least the licensed ActiveX said third set of rules to said third secure control (first protected information. The container, and means for incorporating licensing support code in the ActiveX said third secure container within said control when its developer added licensing second secure container. support to the ActiveX control is the third set of rules. 5 34. A data processing arrangement as in Before an ActiveX control will create a 6 claim 33 further comprising means for copy of itself, the calling application has to applying said third set of rules to govern at pass a license key to the ActiveX control. 7 least one aspect of use of said portion of The license support code in the ActiveX said first protected data. control (third rule set) evaluates the authenticity of the calling application's request. 9 10 35. A data processing arrangement as in Windows Installer operating system service claim 34 further comprising means for enforces the conditional syntax statements 11 applying said second set of rules to govern of the application's signed .msi file. These at least one aspect of use of said portion of statements govern the offer/installation of 12 said first protected data. the ActiveX control 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

### INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP. INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 5.915.019

2	INTERTRUST INFRINGEMENT CHART	
. 3	FOR U.S. PATENT NO. 5,915,019	
. 3	41	Infringing products include all Microsoft
4		tools that support the Microsoft ActiveX
5		licensing model, Visual Studio .NET, the Microsoft Installer SDK, and Operating
٠.		System products that include the Microsoft
. 6	A method comprising performing the	Installer technology.  The signed .msi file created by the ActiveX
7	following steps within a virtual distribution	control developer is the first secure
	environment comprising one or more electronic appliances and a first secure	container. The conditional syntax statement(s) of the ActiveX control
. 8	container, said first secure container	developer's signed .msi file is/are the first
· . 9	comprising (a) a first control set, and	control set.
10	(b) a second secure container comprising a	The first protected information is the
	second control set and first protected information:	ActiveX control.
11	information:	The first alternative for the second secure
12	·	container is the signed and licensed
13		ActiveX control. The second control set is the license support code in the ActiveX
		control.
14		The second alternative for the second
15		secure container is a signed cabinet file
16	·	containing the (signed or unsigned) ActiveX control. The second control set is
10		the license support code in the ActiveX
17	using at least one control from said first	Control.  The ActiveX control developer's
18	control set or said second control set to	conditional syntax statements (first control
19	govern at least one aspect of use of said first protected information while said first	set) in the ActiveX developer's signed .msi file govern the offer/installation of the
19	protected information is contained within	ActiveX control while it is in its signed
20	said first secure container;	.msi file.
21		Alternately, the license support code
		(second control set) in the ActiveX control
22		governs use of the licensed ActiveX control.
23	creating a third secure container	The third secure container is a signed .msi
24	comprising a third control set for governing at least one aspect of use of protected	file. The application developer packages its application in a signed .msi file (third
27	information contained within said third	secure container) and includes conditional
25	secure container;	syntax statements (third control set) in the signed .msi
26	incorporating a first portion of said first	Placing the ActiveX control into the
.	protected information in said third secure	application developer's signed .msi file
27	container, said first portion made up of some or all of said first protected	(third secure container).
28	information; and	
	using at least one control to govern at least	The application developer's conditional

,		
2	one aspect of use of said first portion of said first protected information while said	syntax statement(s) in its signed .msi file govern the offer/installation ActiveX
3	first portion is contained within said third	control while it is in the signed .msi file (third secure container).
4	first secure container further includes a	The second protected information is a second ActiveX control.
5	fourth secure container comprising a fourth	
. 6	control set and second protected information and further comprising the following step:	The first alternative for the fourth secure container is the signed and licensed second ActiveX control. The fourth control set is
7		the license support code in the ActiveX control.
8		1.5
9		The second alternative for the fourth secure container is a signed cabinet file containing
10		the (signed or unsigned) second ActiveX control. The fourth control set is the
11		license support code in the ActiveX control.
. 11	using at least one control from said first	The ActiveX control developer's
12	control set or said fourth control set to	conditional syntax statements (first control
13	govern at least one aspect of use of said second protected information while said	set) in the ActiveX developer's signed .msi file govern the offer/installation of the
13	second protected information is contained	second ActiveX control while it is in its
14	within said first secure container.	signed .msi file.
15		Alternately, the license support code
16		(second control set) in the ActiveX control
•		governs use of the licensed ActiveX
17		control.
18	47. A method as in claim 41, in which said	
	step of creating a third secure container	'
19	includes: creating said third control set by	The application density of the second
20	incorporating at least one control not found	The application developer's conditional syntax statements are not found in either
	in said first control set or said second	the first control set or the second control
.21	control set.	set.
22	52. A method as in claim 41 in which said	
- 1	step of creating a third secure container	
23	occurs at a first site, and further comprising:	
24	copying or transferring said third secure container from said first site to a second	The application developer at first site
25	site located remotely from said first site.	distributes its application to other sites.
26	53. A method as in claim 52 in which said	The application developer at the first site is
77	first site is associated with a content	the content distributor.
. 27	distributor.	
28	54. A method as in claim 53 in which said second site is associated with a user of	The application developer distributes the application to end-users.
.		
•	F	xhibit B
	· · · · · · · · · · · · · · · · · · ·	113

	1	
1	content.	
. 2	EE A made desired in 1 in 54 C at	
. 3		
. 4	said user directly or indirectly initiating communication with said first site.	For Internet downloads, the user initiates the communication with the first site.
. 4		the commencer with the first site.
. 5	64. A method as in claim 54 in which said third control set includes one or more	The application developer's conditional syntax statements (third control set) govern
6	controls at least in part governing the use by said user of at least a portion of said	the installation of the ActiveX control (first protected information).
7	first portion of said first protected information.	
8		
9	76. A method as in claim 41 in which said creation of said third secure container	The third secure container is the application developer's signed .msi file and the third
10	further comprises using a template which specifies one or more of the controls	control set is the conditional syntax statements in that file.
11	contained in said third control set.	Microsoft supplies several template .msi
12		databases for use in authoring installation packages. The UISample msi is the
13		template recommended in the "An Installation Example" on MSDN. This
14		template msi files contains several default conditional syntax statements. At least two
15		of these conditional syntax statements directly govern the installation by blocking progress until the EULA is accepted.
16		
17	78. A method as in claim 52 in which said creation of said third secure container	The third secure container is the application developer's signed .msi file and the third
18	further comprises using a template which specifies one or more of the controls contained in said third control set.	control set is the conditional syntax statements in that file.
19	contained in said tillid control set.	Microsoft supplies several template .msi
20		databases for use in authoring installation packages. The UlSample.msi is the
21		template recommended in the "An Installation Example" on MSDN. This
22		template msi files contains several default conditional syntax statements. At least two
23		of these conditional syntax statements directly govern the installation by blocking
24 ·		progress until the EULA is accepted.
25		
26		
27	•	
28		
ت	·	

### INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP. INTERTRUST INFRINGEMENT CHART

INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 5,915,019

Infringing products include all Microso tools that support the Microsoft Active? licensing model, Visual Studio .NET, the Microsoft Installer SDK, and Operating System products that include the Microsoft Installer technology.    A data processing arrangement comprising: a first secure container comprising first protected information and a first rule set governing use of said first protected information;    The first alternative for the first secure container is the ActiveX control developer's signed .msi file containing a licensed ActiveX control (the first protected information). The conditional syntax statements of the signed .msi file the first rule set.    The second alternative for the first secure container is the signed cabinet file containing the ActiveX control. The license support code in the ActiveX control is the first rule set.    The third alternative for the first secure container is the licensed and signed ActiveX control governed by license support code in the ActiveX control. The second rule set;    The third alternative for the first secure container is the licensed and signed ActiveX control governed by license support code in the ActiveX control. The license support code in the ActiveX control. The second rule set;    The third alternative for the first secure container is the licensed and signed ActiveX control governed by license support code in the ActiveX control. The license support code in the ActiveX control. The second rule set; is the conditional syntax statements of the application. The second rule set is the conditional syntax statements of the application developer's signed .msi file the first rule set.    The third container is a signed cabinet file containing at least the ActiveX control. The licensed ActiveX control. The licensed ActiveX control first rule set.    The third container is a signed cabinet file containing at least the ActiveX control. The licensed ActiveX control. The licensed ActiveX control. The licensed ActiveX control. The licensed ActiveX control. The lic		6. PATENT NO. 5,915,019
a first secure container comprising first protected information and a first rule set governing use of said first protected information;  The first alternative for the first secure container is the ActiveX control developer's signed. msi file containing a licensed ActiveX control (the first rule set.)  The second alternative for the first secure container is the signed cabinet file containing the ActiveX control. The license support code in the ActiveX control is the first rule set.  The third alternative for the first secure container is the signed cabinet file containing is the first rule set.  The third alternative for the first secure container is the license support code in the ActiveX control. The license support code in the ActiveX control.  The second secure container is the signed cabinet file containing at least and third rule set governing at least a portion of said first protected information and a third rule set governing use of said portion of said first protected information as said second secure container, said means for copying or transferring at least a portion of said first protected information is said second secure container, said means for copying or transferring comprising:  means for incorporating said third secure container. The licensing support code in the ActiveX control (first protected information is a signed cabinet file containing at least the ActiveX control is third rule set governing use of said portion of said first protected information is a signed cabinet file containing at least the ActiveX control (first protected information is a signed cabinet file containing at least the ActiveX control (first protected information is a signed cabinet file containing at least the ActiveX control (first protected information is a signed cabinet file containing at least the ActiveX control (first protected information is a signed cabinet file containing at least the ActiveX control is third rule set.  A data processing arrangement as in licensed and signed action are secure container.  Th	81.	Infringing products include all Microsoft tools that support the Microsoft ActiveX licensing model, Visual Studio .NET, the Microsoft Installer SDK, and Operating System products that include the Microso Installer technology.
container is the ActiveX control developer's signed .msi file containing a licensed ActiveX control. The conditional syntax statements of the signed cabinet file containing the ActiveX control. The license support code in the ActiveX control is the first rule set.  The second alternative for the first secure container is the signed cabinet file containing the ActiveX control. The license support code in the ActiveX control is the first rule set.  The third alternative for the first secure container is the signed cabinet file containing the ActiveX control. The license support code in the ActiveX control. The second secure container comprising a second rule set;  The third alternative for the first secure container is the signed cabinet file containing the ActiveX control. The second secure container is the signed active X control. The second secure container is the signed cabinet file containing at least application developer active is the conditional syntax statements of the application developer active is the conditional syntax statements of the application developer active is the conditional syntax statements of the application developer active is the conditional syntax statements of the application developer active is the conditional syntax statements of the application developer active is the conditional syntax statements of the application developer active is the conditional syntax statements of the application developer active is the conditional syntax statements of the application developer active is the conditional syntax statements of the application developer active is the conditional syntax statements of the application developer active is the conditional syntax statements of the application developer active is the conditional syntax statements of the application developer active is the conditional syntax statements of the application developer active is the conditional syntax statements of the application developer active is the conditional syntax statements of the application developer active is t	A data processing arrangement comprising	
The second alternative for the first secure container is the signed cabinet file containing the ActiveX control. The license support code in the ActiveX control is the first rule set.  The third alternative for the first secure container is the licensed and signed ActiveX control governed by license support code in the ActiveX control.  The second secure container is the signed activeX control governed by license support code in the ActiveX control.  The second secure container is the signed activeX control.  The second secure container is the signed activeX control.  The second secure container is the signed activeX control.  The second secure container is the signed activeX control.  The second secure container is the signed activeX control.  The second secure container is the signed activeX control.  The second secure container is the signed activeX control.  The second secure container is the signed activeX control.  The second secure container is the signed activeX control.  The second secure container is the signed activeX control.  The second secure container is the signed activeX control.  The second secure container is the signed activeX control.  The second secure container is the signed activeX control.  The second secure container is the signed activeX control.  The second secure container is the signed activeX control.  The third containing at least the ActiveX control (first protected information) in a signed cabinet file (third secure container). The licensing support code in the ActiveX control is third secure container, said means for copying or transferring comprising:  means for incorporating said third secure container, said means for applying at least one rule from add third rule set to at least in part govern to act the signed activeX control is third rule set to at least in part govern to act the signed activeX control is third rule set to at least the activeX control (first protected information) in a signed cabinet file containing at least the ActiveX control is the secure container.  P	protected information and a first rule set governing use of said first protected	container is the ActiveX control developer's signed .msi file containing a licensed ActiveX control (the first protected information). The conditional syntax statements of the signed .msi file at
container is the signed cabinet file containing the ActiveX control. The license support code in the ActiveX control is the first rule set.  The third alternative for the first secure container is the licensed and signed ActiveX control governed by license support code in the ActiveX control. The second rule set;  The third alternative for the first secure container is the licensed and signed ActiveX control governed by license support code in the ActiveX control. The second secure container as the conditional syntax statements of the application developer package its application. The second rule set is the conditional syntax statements of the application developer's signed .msi file which the application developer's signed .msi file third containing at least the ActiveX control. Putting the licensed ActiveX control (first protected information to said first protected information us as a signed cabinet file (third secure container). The licensing support code in the ActiveX control.  Putting the licensed ActiveX control (first protected information) in a signed cabinet file (third secure container). The licensing support code in the ActiveX control.  Putting the licensed ActiveX control (first protected information) in a signed cabinet file (third secure container). The licensing support code in the ActiveX control.  Putting the licensed ActiveX control (first protected information) in a signed cabinet file (third secure container). The licensing support code in the ActiveX control (first protected information) as a signed cabinet file (third secure container). The licensing support code in the ActiveX control (first protected information) are application developer's signed .msi file (third secure container). The licensing support code in the ActiveX control (first protected information) are application developer's signed .msi file (third secure container). The licensing support code in the ActiveX control (first protected information) are application developer's signed .msi file (third secure container). The l		the mst fulle set.
container is the licensed and signed ActiveX control governed by license support code in the ActiveX control.  The second secure container comprising a second rule set;  The second secure container is the signed means for creating and storing a third secure container; and means for copying or transferring at least a portion of said first protected information and a third rule set governing use of said portion of said first protected information to said second secure container, said means for incorporating said third secure container within said second secure container within said second secure container.  The third container is the licensed and signed ActiveX control.  The second secure container is the licensed in the ActiveX control.  The third container is a signed cabinet file containing at least the ActiveX control.  Putting the licensed ActiveX control (first protected information) in a signed cabinet file (third secure container). The licensing support code in the ActiveX control is third secure container, said means for copying or transferring comprising:  means for incorporating said third secure container within said second secure container.  The third container is the licensed and signed in the ActiveX control.  Putting the licensed ActiveX control (first protected information) in a signed cabinet file (third secure container). The licensing support code in the ActiveX control is third secure container.  Packaging the signed cabinet file in the signed in the ActiveX control is third rule set ensures the user is licensed.  The third container is the signed in the set of the application developer package its application. The second rule set is the conditional syntax statements of the application developer package its application. The second rule set is the conditional syntax statements of the application developer package its application. The second rule set is the conditional syntax statements of the application developer package its application. The second rule set is the conditional syntax statement		containing the ActiveX control. The license support code in the ActiveX control
container is the licensed and signed ActiveX control governed by license support code in the ActiveX control.  The second secure container comprising a second rule set;  The second secure container is the signed means for creating and storing a third secure container; and means for copying or transferring at least a portion of said first protected information and a third rule set governing use of said portion of said first protected information to said second secure container, said means for incorporating said third secure container within said second secure container within said second secure container.  The third container is the licensed and signed ActiveX control.  The second secure container is the licensed in the ActiveX control.  The third container is a signed cabinet file containing at least the ActiveX control.  Putting the licensed ActiveX control (first protected information) in a signed cabinet file (third secure container). The licensing support code in the ActiveX control is third secure container, said means for copying or transferring comprising:  means for incorporating said third secure container within said second secure container.  The third container is the licensed and signed in the ActiveX control.  Putting the licensed ActiveX control (first protected information) in a signed cabinet file (third secure container). The licensing support code in the ActiveX control is third secure container.  Packaging the signed cabinet file in the signed in the ActiveX control is third rule set ensures the user is licensed.  The third container is the signed in the set of the application developer package its application. The second rule set is the conditional syntax statements of the application developer package its application. The second rule set is the conditional syntax statements of the application developer package its application. The second rule set is the conditional syntax statements of the application developer package its application. The second rule set is the conditional syntax statement	·	The third alternative for the first
The second secure container is the signed second rule set;  means for creating and storing a third secure container; and means for copying or transferring at least a portion of said first protected information and a third rule set governing use of said portion of said first protected information to said second secure container, said means for copying or transferring comprising:  means for incorporating said third secure container within said second secure container within said second secure container.  The second secure container is the signed set is the conditional syntax statements of the application developer's signed signed is application. The second rule set is the conditional syntax statements of the application developer's signed signed is application. The second rule set is the conditional syntax statements of the application developer's signed signed is application. The second rule set is the conditional syntax statements of the application developer's signed signed is application. The second rule set is the conditional syntax statements of the application developer's signed signed is application. The second rule set is the conditional syntax statements of the application developer's signed signed is application. The second rule set is the conditional syntax statements of the application developer's signed signed is application. The second rule set is the conditional syntax statements of the application developer's signed signed is application. The second rule set is the conditional syntax statements of the application developer's signed signed is application.  Putting the licensed ActiveX control (first protected information) in a signed cabinet file (third secure container). The licensing support code in the ActiveX control is third secure container.  Packaging the signed cabinet file on the ActiveX control is third secure container.  The third container is a signed cabinet file on the ActiveX control.  Packaging the signed said the ActiveX control is the support code in the ActiveX control is the ap		container is the licensed and signed ActiveX control governed by license
The third container is a signed cabinet file containing at least a portion of said first protected information and a third rule set governing use of said portion of said first protected information to said second secure container, said means for copying or transferring comprising:  means for incorporating said third secure container within said second secure container within said second secure container.  22. A data processing arrangement as in claim 81 further comprising:  means for applying at least one rule from aid third rule set to at least in part govern t least one factor related to use of said cortion of said first protected information.  The third container is a signed cabinet file containing at least the ActiveX control.  Putting the licensed ActiveX control (first protected information) in a signed cabinet file (third secure container). The licensing support code in the ActiveX control is third rule set.  Packaging the signed cabinet file containing at least the ActiveX control.  Putting the licensed ActiveX control (first protected information) in a signed cabinet file (third secure container). The licensing support code in the ActiveX control is third rule set.  Packaging the signed cabinet file (third secure container). The licensing support code in the ActiveX control is third rule set.  The third containing at least the ActiveX control (first protected information) in a signed cabinet file (third secure container). The licensing support code in the ActiveX control is third rule set.  The third containing at least the ActiveX control (first protected information) in a signed cabinet file (third secure container).  The third rule set set of a least the ActiveX control (first protected information) in a signed cabinet file (third secure container).  The third rule set set of a least the ActiveX control (first protected information) in a signed cabinet file (third secure container).  The third rule set set of a least the ActiveX control (first protected information) in a signed cabinet file (third s	a second secure container comprising a second rule set;	The second secure container is the signed .msi file which the application developer package its application. The second rule set is the conditional syntax statements of
means for copying or transferring at least a portion of said first protected information and a third rule set governing use of said portion of said first protected information to said second secure container, said means for copying or transferring comprising:  means for incorporating said third secure container within said second secure container.  C2. A data processing arrangement as in claim 81 further comprising:  means for applying at least one rule from aid third rule set to at least in part govern t least one factor related to use of said ortion of said first protected information.  C3. A data processing arrangement as in claim 81 further comprising:  means for applying at least one rule from aid third rule set to at least in part govern t least one factor related to use of said ortion of said first protected information.  C3. A data processing arrangement as in containing at least the ActiveX control.  Putting the licensed ActiveX control.  Futting the licensed ActiveX control.	means for creating and storing a third	The third container is a signed cabinet file
portion of said first protected information and a third rule set governing use of said portion of said first protected information to said second secure container, said means for copying or transferring comprising:  means for incorporating said third secure container within said second secure container within said second secure container.  Putting the licensed ActiveX control (first protected information) in a signed cabinet file (third secure container). The licensing support code in the ActiveX control is third rule set.  Packaging the signed cabinet file in the signed .msi file.  Packaging the signed cabinet file in the signed .msi file.  The third rule set ensures the user is licensed.  The third rule set ensures the user is licensed.	secure container; and	COMBINING ST least the ActiveV control
support code in the ActiveX control is this rule set.  support code in the ActiveX control is this rule set.  support code in the ActiveX control is this rule set.  support code in the ActiveX control is this rule set.  Packaging the signed cabinet file in the signed .msi file.  Packaging the signed cabinet file in the signed .msi file.  Packaging the signed cabinet file in the signed .msi file.  The third rule set ensures the user is licensed.  The third rule set ensures the user is licensed.  A data processing arrangement as in licensed.	portion of said first protected information	Putting the licensed ActiveX control (first protected information) in a signed cabinet
rule set.  rule set.  rule set.  rule set.  rule set.  rule set.  rule set.  rule set.  rule set.  rule set.  Packaging the signed cabinet file in the signed .msi file.  Packaging the signed cabinet file in the signed .msi file.  Packaging the signed cabinet file in the signed .msi file.  Packaging the signed cabinet file in the signed .msi file.  The third rule set ensures the user is licensed.  The third rule set ensures the user is licensed.  The third rule set ensures the user is licensed.  The third rule set ensures the user is licensed.	portion of said first protected information	SIIDDOTE code in the ActiveV and an in the
means for incorporating said third secure container within said second secure container.  Packaging the signed cabinet file in the signed .msi file.  Packaging the signed cabinet file in the signed .msi file.  Packaging the signed cabinet file in the signed .msi file.  The third rule set ensures the user is licensed.  The third rule set ensures the user is licensed.  The third rule set ensures the user is licensed.  The third rule set ensures the user is licensed.	to said second secure container, said means	s rule set.
secure container within said second signed .msi file.  32. A data processing arrangement as in claim 81 further comprising: neans for applying at least one rule from aid third rule set to at least in part govern t least one factor related to use of said contion of said first protected information.  3. A data processing arrangement as in	means for incorporating said third	Packaging the signal 1: Ct.
32. A data processing arrangement as in claim 81 further comprising: neans for applying at least one rule from aid third rule set to at least in part govern t least one factor related to use of said contion of said first protected information.  The third rule set ensures the user is licensed.  3. A data processing arrangement as in	secure container within said second	signed .msi file
reans for applying at least one rule from aid third rule set to at least in part govern t least one factor related to use of said contion of said first protected information.  The third rule set ensures the user is licensed.  It is a set of third rule set ensures the user is licensed.  The third rule set ensures the user is licensed.	secure container.	100000000000000000000000000000000000000
reans for applying at least one rule from aid third rule set to at least in part govern t least one factor related to use of said contion of said first protected information.  The third rule set ensures the user is licensed.  It is a set of third rule set ensures the user is licensed.  The third rule set ensures the user is licensed.	2 A data programa	
aid third rule set to at least in part govern t least one factor related to use of said contion of said first protected information.  3. A data processing arrangement as in	laim 81 further comprising:	
1 least one factor related to use of said ontion of said first protected information.  3. A data processing arrangement as in	aid third rule set to at least in part govern	The third rule set ensures the user is
3. A data processing аттапдетелт as in	t least one factor related to use of said	
iain 62 iurner comprising:	3. A data processing arrangement as in	
	ann 82 further comprising:	

Exhibit B

	i
	2
•	3
	4
	5 6 7
(	6
•	7
	3
. 9	,
10	)
- 11	
12	- 11
13	
14	- 11
15	
16	
17	
18	
19	
20	
. 21	
22	
23	
24	
25	
26	
	11

means for applying at least one rule from said second rule set to at least in part govern at least one factor related to use of said portion of said first protected information.

The second rule set governs the offer/installation of first protected information.

Exhibit B

27

# INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP. INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 5,915,019

3	FOR U.S. PATENT NO. 5,915,019	
5		Infringing products include all Microsoft tools that support the Microsoft ActiveX licensing model, Visual Studio .NET, the Microsoft Installer SDK, and Operating System products that include the Microsoft Installer technology.
.7	A method comprising the following steps: creating a first secure container comprising	The first and total in first and in first an
. 8		The first protected information is the ActiveX control.
9	• • • • • • • • • • • • • • • • • • • •	The first alternative for the first secure
10		ActiveX control. The first rule set is the license support code in the ActiveX.
11	·	control.
12		The second alternative for the first secure container is an (signed or unsigned)
13		ActiveX control with license support
14		contained within a signed cabinet file. The first rule set is the ActiveX license support code.
15	storing said first secure container in a first memory;	The first secure container is stored at the ActiveX control developer's location.
16	creating a second secure container	The second secure container is the
17	comprising a second rule set;	application developer's signed .msi file. The conditional syntax statements of the signed .msi file are the second rule set.
18	storing said second secure container in a second memory;	The second secure container is stored at the application developer's location.
19	copying or transferring at least a first portion of said first protected information	The ActiveX control developer packages the control in a signed .msi file for
20	to said second secure container, said copying or transferring step comprising:	distribution to the application developer's site.
21	creating a third secure container	The third secure container is the ActiveX
22	comprising a third rule set;	control developer's signed .msi file containing a licensed ActiveX control. The
23	·	conditional syntax statements of the signed msi file are the third rule set.
24	copying said first portion of said first protected information;	In preparation for using a msi authoring
		tool, such as Microsoft's Orca, copying the ActiveX control to a package staging area.
25	transferring said copied first portion of said first protected information to	Using msi authoring tool to import the
26	said third secure container; and	control into the signed .msi file.
27	copying or transferring said copied first portion of said first protected	The application developer installs the ActiveX control, which involves removing
28	information from said third secure	it from the ActiveX developer's signed
20	container to said second secure container.	msi file and installing it into its environment. Subsequently, the
- t		!!

Exhibit B

293482.02

1

		. · · · · · · · · · · · · · · · · · · ·
•		
1		application developer places the ActiveX
2		control into its signed .msi file when it is packaging its application.
3	87. A method as in claim 85 in which said	The entire ActiveV control is social
4	copied first portion of said first protected	The entire ActiveX control is copied.
5	information consists of the entirety of said first protected information.	
3		
6	89. A method as in claim 85 in which	
7	said first memory is located at a first site,	The first memory is located at the ActiveX control developer's site.
·	said second memory is located at a second site remote from said first site, and	The second memory is located at the
8.	said step of copying or transferring said	application developer's site.  The ActiveX control developer's signed
9	first portion of said first protected information to said second secure container	msi file is transferred from its site to the
10	further comprises copying or transferring	site of the application developer.
	said third secure container from said first	,
11	site to said second site.	
12		
13		•
14		•
15		
l		
16	·	
· II		· · · · · · · · · · · · · · · · · · ·

Exhibit B

### INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.

INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 5,915,019

	3		· ·
	. 4	85. (alternate infringing scenario)	Infringing products include all Microsoft tools that support the Microsoft ActiveX
	5		licensing model, Visual Studio .NET, the Microsoft Installer SDK, and Operating
•	6		System products that include the Microsoft Installer technology.
	7	A method comprising the following steps:	• • • • • • • • • • • • • • • • • • • •
	8	creating a first secure container comprising a first rule set and first protected information;	The first protected information is the ActiveX control.
	. 9	meormation,	The first alternative for the first secure
			container is the signed and licensed
	10		ActiveX control. The first rule set is the license support code in the ActiveX
	11		control.
	12		The second alternative for the first secure container is a (signed or unsigned) ActiveX
	13		control with license support contained
	13		within a signed cabinet file. The first rule
	14		set would remain the ActiveX license support code.
	15		The third alternative for the first secure
	16		container is a signed msi file in which the
•	10		ActiveX control developer packaged its
	17		ActiveX control. The first rule set is the
			conditional syntax statement(s) of the signed msi file.
	18	storing said first secure container in a first	The first secure container is stored at the
	19	memory;	ActiveX control developer's location.
		creating a second secure container	The second secure container is the
	20	comprising a second rule set;	application developer's signed .msi file. The conditional syntax statements of the
	21		signed .msi file are the second rule set.
		storing said second secure container in a	The second secure container is stored at the
	22	second memory;	application developer's location.
	22	copying or transferring at least a first portion of said first protected information	The ActiveX control is placed in a cabinet file signed by the application developer and
	23	to said second secure container, said	the signed cabinet file is placed in a .msi
٠	24	copying or transferring step comprising:	file signed by the application developer.
		creating a third secure container	The third secure container is signed cabinet
	25	comprising a third rule set;	file in which the application developer
	26		placed licensed ActiveX. The third rule set is the license support code in the ActiveX
	26		control.
	27	copying said first portion of said	Copying ActiveX control.
	ł	first protected information;	
	28	transferring said copied first portion of said first protected information to	Transferring ActiveX control to signed cabinet file.
	Į.		

Exhibit B

2

•		
1	said third secure container; and	
2	copying or transferring said copied first portion of said first protected	The application developer places the signed cabinet file into its signed .msi file when it
3	information from said third secure container to said second secure container.	is packaging its application.
4	container.	<u> </u>
5	87. A method as in claim 85 in which said copied first portion of said first protected information consists of the entirety of said	The entire ActiveX control is copied.
. 6	first protected information.	
. 7	O2 A method as in alaim 95 in maint	
8	93. A method as in claim 85 in which said step of copying transferring said	The ActiveX control is placed in a cabinet
9	copied first portion of said first protected information from said third secure	file signed by the application developer and the signed cabinet file is placed in a .msi
. 10	container to said second secure container further comprises storing said third secure container in said second secure container.	file signed by the application developer.
11		
12	·	
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		·
25		
26		•
27		

### INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP. INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 5,915,019

,		
. 4	I.	Infringing products include the .NET Framework SDK, Microsoft Visual Studio
5		NET, the Microsoft Installer SDK, and products that include the Microsoft NET
6		CLR, and the Microsoft Installer technology.
. 7	A method of operating on a first secure container arrangement having a first set of	The first protected content is a signed and licensed .NET component used by the
- 8	controls associated therewith, said first secure container arrangement at least in	.NET assembly. The .NET assembly is distributed with a signed and governed .msi
9	part comprising a first protected content file, said method comprising the following	file. The second protected content is another signed and licensed .NET
10		component that is used by the .NET assembly.
- 11	one electronic appliance:	
12	using at least one control associated with said first secure container arrangement for governing, at least in part, at least one	The first protected content is signed and licensed .NET component (first secure container) contained within the .NET
13	aspect of use of said first protected content file while said first protected content file is	assembly. The one control is a declarative statement(s) within the assembly's header.
. 14	contained in said first secure container arrangement;	balloni, wall are assembly a header.
15	creating a second secure container arrangement having a second set of	The protected content is the same as the first protected content plus the additional
16	controls associated therewith, said second set of controls governing, at least in part, at	implementation information included in the signed .msi file. The second secure
-17	least one aspect of use of any protected content file contained within said second	container is the signed .msi file created for the .NET assembly. The signed .msi file's
18	secure container arrangement;	conditional syntax statements are the second set of controls that control the
19		offer/installation of the .NET assembly.
20	transferring at least a portion of said first protected content file to said second secure container arrangement, said portion made	The entire .NET assembly is included in the signed .msi file.
21	up of at least some of said first protected content file; and	Packaging the .NET assembly in the signed .msi file involves the following process
22		steps. In preparation for using a msi
23		authoring tool, such as Microsoft's Orca, copying the .NET component to a package
24		staging area. Using msi authoring tool to import the .NET component into the signed .msi file.
25	using at least one rule to govern at least one	The conditional syntax statement(s) of the
26	aspect of use of said first protected content file portion while said portion is contained	signed .msi file (second secure container) control(s) the offer/installation of the .NET
27	within said second secure container arrangement:	assembly.
	in which	
28	said first secure container arrangement	The first alternative for the third secure
ļ	comprises a third secure container	container is a licensed and signed .NET
		CH+

Exhibit B

293482.02

1

2	arrangement comprising a third set of controls and said first protected content file, and	component governed by the set of declarative statements comprising the LicenseProviderAttribute (third set of controls).
4 5		The second alternative for the third secure container is a .NET component whose hash is included in the header of the .NET
6		assembly. The set of declarative statements comprising the LicenseProviderAttribute is the third set of
7	said first secure container arrangement	controls.  The first alternative for the fourth secure
. 8	further comprises a fourth secure container arrangement comprising a fourth set of controls and a second protected content	container is another licensed and signed .NET component governed by the set of
9	file.	declarative statements comprising the LicenseProviderAttribute (fourth set of
. 10		controls).
. 11		The second alternative for the fourth secure container is the container created when the
12		hash of the .NET component is included in the header information of the .NET
. 13		assembly. The set of declarative statements comprising the
-14		LicenseProviderAttribute is the fourth set of controls.
15		
16	·	
17		
18		
19		
20		•
21		
22		
23		
24		
25		•
26		
27		
Į.		

	INTERTRUST INFRINGEMENT CHART	
2	FOR U.S. PATENT NO. 5,915,019	
• 3	3	
	33.	Infringing products include the .NET
. 4		Framework SDK, Microsoft Visual Studio NET, the Microsoft Installer SDK, and
5		products that include the Microsoft .NET
6	. 1	CLR, and the Microsoft Installer technology.
C	A data processing arrangement comprising	The first protected information is the .NET
7	at least one storing arrangement that at least temporarily stores a first secure	component.
8	container comprising first protected data	The first alternate for the first secure
9	and a first set of rules governing use of said first protected data,	container is the signed .msi file in which the .NET component developer packaged
		its .NET component. The first set of rules
- 10		is the conditional syntax statements of the signed .msi file.
. 11		
12		The second alternative for the first secure container is a licensed and signed .NET
12		component governed by the set of
13		declarative statements comprising the
14		LicenseProviderAttribute of the .NET component (first set of controls).
•		
15		The third alternative for the first container is a signed cabinet file containing a (signed
16		or unsigned) .NET component with license
17		support. The first set of controls is the set of declarative statements comprising the
		LicenseProviderAttribute of the .NET
18		component.
19	and at least temporarily stores a second	The second protected data is the .NET
20	secure container comprising second protected data different from said first	assembly developer's assembly that includes/uses the .NET component.
	protected data and a second set of rules	
21	governing use of said second protected data; and	The first alternative for the second secure container is a signed .msi file in which the
22		.NET assembly developer packaged its
23		multi-file assembly (second protected data). The second set of rules is the
	İ	conditional syntax statements of the signed
24		.msi file that governs the offer/installation of the .NET assembly.
25.		
26		The second alternative for the second secure container is a signed .NET
. 20		assembly. The second set of rules is the
27		declarative rules within the assembly's header.
28	a data transfer arrangement, coupled to at	The third secure container is a signed .NET.
		assembly governed by declarative rules in
	·	# · · · · · · · · · · · · · · · · · · ·

Exhibit B

transferring at least a portion of said first protected data and a third set of rules governing use of said portion of said first protected data to said second secure container,  transferring at least a portion of said first protected data and a third set of rules alternative third rule set is the declarative statements comprible License Provider Attribute. The assembly includes the .NET of the secure .NET assembly is signed .msi file (second secure container created by hashing to component and including the header information of a .NET	e set of ising the e .NET component. included in a e container). ntainer is the
protected data and a third set of rules governing use of said portion of said first protected data to said second secure container,  alternative third rule set is the declarative statements compri LicenseProviderAttribute. The assembly includes the .NET of The secure .NET assembly is signed .msi file (second secure container created by hashing to component and including the	e set of ising the e .NET component. included in a e container). ntainer is the
governing use of said portion of said first protected data to said second secure container,  declarative statements compring LicenseProviderAttribute. The assembly includes the .NET of The secure .NET assembly is signed .msi file (second secure container created by hashing to component and including the	ising the e.NET component. included in a e container). ntainer is the
protected data to said second secure container,  LicenseProviderAttribute. The assembly includes the .NET of The secure .NET assembly is signed .msi file (second secure container created by hashing to component and including the	e .NET component. included in a e container). ntainer is the
container,  assembly includes the .NET of The secure .NET assembly is signed .msi file (second secure container created by hashing to component and including the	component. included in a e container). ntainer is the
The secure .NET assembly is signed .msi file (second secure container created by hashing to component and including the	included in a e container).
signed .msi file (second secure container created by hashing the	e container).
An alternative third secure container created by hashing to component and including the	ntainer is the
An alternative third secure concontainer created by hashing to component and including the	
container created by hashing to component and including the	
component and including the	
7   header information of a NET	
A Same	assembly.
The .NET component is include	
signed and governed .NET as	sembly
(second secure container). The	ne third set of
rules is the set of declarative s	
comprising the LicenseProvid	erAttribute.
10	
An alternative third secure con	
signed cabinet file containing	
component and which is desting	
signed .msi file (second secure	
The third set of rules is the set	
declarative statements compris	sing the
LicenseProviderAttribute.	
14   further comprising	
means for creating and storing, in said at The first alternative for the thi	
least one storing arrangement, a third container is a signed .NET ass	
secure container; this case, the second secure co	intainer is the
signed .msi file.	
The second alternative for the	داداه
<b>*/</b> 11	
container is the container creat including a hash of the .NET c	
18 Including a hash of the .NET c	
In this case, the second secure	container is
either the signed .msi file or th .NET assembly.	ie signed
.NEI assembly.	1
The third alternative for the thi	ird container
21   I he third alternative for the third is a cabinet file signed by the	
	the NET
assembly developer containing assembly and/or the .NET com	popent In
this case the signed .msi file is secure container.	me second
	rd convec
27	
container, and means for incorporating information). The third set of a	
said third secure container within said declarative rule within the .NE	
second secure container. assembly's header. The .NET	
placed in a signed .msi file (see	cond secure
container).	
	•

	•	
2 3 4 5		The second alternative for the third secure container is the container that results when the hash of the .NET component is added to the .NET assembly header information. The third set of rules is the set of declarative statements comprising the LicenseProviderAttribute added to the assembly.
6		The third alternative for the third secure container is a cabinet file signed by the
7		NET assembly developer containing the NET assembly and/or the .NET
8		component. The third set of rules is a declarative rule(s) within the .NET
9		assembly's header and/or the set of declarative statements comprising the
10		LicenseProviderAttribute added to the assembly
11	24.4.14	
12	34. A data processing arrangement as in claim 33 further comprising means for applying said third set of rules to govern at	When the third rule set is the declarative statement(s) of the assembly header, the runtime CLR enforces the statements.
13	least one aspect of use of said portion of said first protected data.	When the third set of rules is the set of
14	said inst protected data.	declarative statements comprising the LicenseProviderAttribute added to the
. 15		assembly, the license support code in the .NET component evaluates the authenticity
16		of the calling assembly's request.
17	35. A data processing arrangement as in claim 34 further comprising means for	When the second set of rules is the conditional syntax statements of the signed
18	applying said second set of rules to govern at least one aspect of use of said portion of	.msi file, the Windows Installer operating system service enforces the conditional
19	said first protected data.	syntax statements of .NET assembly's signed .msi file, which govern the
20		offer/installation of the .NET component.
21		When the second set of rules is the declarative statement(s) within the
22		assembly's header, the runtime CLR enforces the statements.
23		
24		
-25		÷
26		
27		
28		
٠.	•	

2

### INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP. INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 5.915.019

-	FOR U.S. PA	TENT NO. 5,915,019
3	41.	Infringing products include the .NET Framework SDK, Microsoft Visual Studio
4		.NET, the Microsoft Installer SDK, and
5		products that include the Microsoft .NET CLR, and the Microsoft Installer
6		technology.
7	A method comprising performing the following steps within a virtual distribution environment comprising one or more	The signed .msi file created by the .NET component developer is the first secure container. The first conditional syntax
8 9	electronic appliances and a first secure container, said first secure container comprising (a) a first control set, and	statement(s) of the .NET component developer's signed .msi file is/are the first control set.
-		
10	(b) a second secure container comprising a second control set and first protected information:	The first protected information is the .NET component.
11		The first alternative for the second secure
12		container is the signed and licensed .NET component. The second control set is the
13		set of declarative statements comprising the LicenseProviderAttribute.
14		The second alternative for the second
15	·	secure container is a signed cabinet file. The second control set remains the set of
16		declarative statements comprising the LicenseProviderAttribute.
17	using at least one control from said first control set or said second control set to	The .NET component developer's conditional syntax statements (first control
18	govern at least one aspect of use of said	set) in its signed .msi file governs the
19	first protected information while said first protected information is contained within	offer/installation of the .NET component while it is in the signed .msi file.
20	said first secure container;	Alternately, the set of declarative
20		statements comprising the
21		LicenseProviderAttribute (second control set) of the licensed .NET component
22	creating a third secure container	governs use of the .NET component.  The first alternative for the third secure
23	creating a third secure container comprising a third control set for governing	container is a signed .NET assembly, the
24	at least one aspect of use of protected information contained within said third	protected information is the .NET component and the third control set is the
į	secure container;	declarative statement(s) within the .NET
25		assembly's header.
26		The second alternative for the third secure container is a signed .msi file in which the
27		.NET assembly developer packages its
28		.NET assembly and the third control set is the conditional syntax statement(s) in the signed .msi file.
H		

1	incorporating a first portion of said first	In the first alternative, placing the .NET
2	protected information in said third secure container, said first portion made up of	component into the signed .NET assembly.
3	some or all of said first protected information; and	In the second alternative, placing the .NET component into the. Net assembly
4		developer's signed msi file.
5	using at least one control to govern at least one aspect of use of said first portion of	In the first alternative, the .NET assembly developer's declarative statement(s) within
6	said first protected information while said first portion is contained within said third	the .NET assembly's header govern(s) the use of the .NET component while it is in
. 7	secure container.	the signed .NET assembly.
		In the second alternative, the conditional syntax statements of the .NET assembly
8		developer's signed .msi file govern the
.9		offer/installation of the .NET component while it is in the signed .msi file.
10	42. A method as in claim 41, in which said	The second protected information is;a
11	first secure container further includes a fourth secure container comprising a fourth	second .NET component.
12	control set and second protected	The first alternative for the fourth secure
13	information and further comprising the following step:	container is the signed and licensed second .NET component. The fourth control set is
14		the set of declarative statements comprising the LicenseProviderAttribute of the second
1.5		.NET component.
15		The second alternative for the fourth secure container is a second signed cabinet file.
16		The fourth control set is the set of
17		declarative statements comprising the LicenseProviderAttribute.
18	using at least one control from said first control set or said fourth control set to	The .NET component developer's conditional syntax statements (first control
19	govern at least one aspect of use of said second protected information while said	set) in its signed .msi file governs the offer/installation of the second .NET
20	second protected information is contained	component while it is in the signed .msi file.
21	within said first secure container.	
22		Alternately, the set of declarative statements comprising the
		LicenseProviderAttribute (fourth control set) of the licensed second .NET
23		component governs use of the second .NET
24		component.
25	47. A method as in claim 41, in which said step of creating a third secure container	
26	includes:	TI NET
27	creating said third control set by incorporating at least one control not found	The .NET assembly developer's declarative statements (first alternative for third control
.	in said first control set or said second control set.	set) and/or the developer's conditional syntax statements (second alternative for
28	Condo Sct.	the third control set) are not found in either

•		
1		the first control set or the second control set.
2		1000
3	step of creating a third secure container	:
4	comprising:	
5	copying or transferring said third secure container from said first site to a second site located remotely from said first site.	The .NET assembly developer at first site distributes its assembly to other sites.
6	she located remotely from said first she.	
7 8	53. A method as in claim 52 in which said first site is associated with a content distributor.	The .NET assembly developer's business module is used to create and distribute its assembly.
0		
9	54. A method as in claim 53 in which said second site is associated with a user of content.	The .NET assembly developer distributes the assembly to end-users.
-		
11	55. A method as in claim 54 further comprising the following step:	
12	said user directly or indirectly initiating communication with said first site.	For Internet downloads, the user initiates the communication with the first site.
13	Communication with said first site.	the communication with the first site.
14	64. A method as in claim 54 in which said third control set includes one or more	When the third control set is the .NET assembly developer's declarative
15	controls at least in part governing the use by said user of at least a portion of said	statement(s) within the .NET assembly's header, it governs the user's use of the
16	first portion of said first protected information.	NET component (first protected information).
17		When the third control set is the .NET assembly developer's conditional syntax
18		statements of the .NET assembly developer's signed .msi file, it governs the
19		user's offer acceptance/installation of the .NET component (first protected
20		information).
21	76. A method as in claim 41 in which said creation of said third secure container	When the third secure container is the .NET assembly developer's signed .msi file
22	further comprises using a template which	and the third control set is the conditional
23	specifies one or more of the controls contained in said third control set.	syntax statements in that file.
24		Microsoft supplies several template .msi databases for use in authoring installation
25		packages. The UISample.msi is the template recommended in the "An
26		Installation Example" on MSDN. This template msi files contains several default
27		conditional syntax statements. At least two of these conditional syntax statements
28		directly govern the installation by blocking progress until the EULA is accepted.

78. A method as in claim 52 in which said creation of said third secure container further comprises using a template which specifies one or more of the controls contained in said third control set.

When the third secure container is the .NET assembly developer's signed .msi file and the third control set is the conditional syntax statements in that file.

Microsoft supplies several template .msi databases for use in authoring installation packages. The UISample.msi is the template recommended in the "An Installation Example" on MSDN. This template msi files contains several default conditional syntax statements. At least two of these conditional syntax statements directly govern the installation by blocking progress until the EULA is accepted.

INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 5,915,019

	FOR U.S. PATENT NO. 5,915,019	
3	81.	Infringing products include the .NET
4		Framework SDK, Microsoft Visual Studio
7		.NET, the Microsoft Installer SDK, and
5		products that include the Microsoft .NET
	,	CLR, and the Microsoft Installer
6	A data processing arrangement comprising:	technology.
7	A data processing arrangement comprising:  a first secure container comprising first	The first protected information is the .NET
ŀ	protected information and a first rule set	component.
8	governing use of said first protected	
	information;	The first alternative for the first secure
9		container is the signed .msi file in which
10		the .NET component developer packaged its assembly. The first rule set is the
10		conditional syntax statements written by
11	·	the .NET component developer and placed
		into the signed .msi file.
12		The second alternative for the first secure
13		container is the signed cabinet file
13		containing the (signed or unsigned) .NET
14		component. The set of declarative
		statements comprising the
15	·	LicenseProviderAttribute when its developer added licensing support to the
16		assembly is the first rule set.
10	. '	
17		The third alternative for the first secure
		container is the licensed and signed .NET
1.8		component governed by the set of declarative statements comprising the
19		LicenseProviderAttribute (first rule set)
		added by the .NET component developer.
20	a second secure container comprising a	The first alternative for the second secure
۸, ا	second rule set;	container is the signed .msi file in which
21	:	the .NET assembly developer packaged its .NET assembly. The second rule set is the
22	·	conditional syntax statements written by
		the .NET assembly developer and placed
23	·	into the signed .msi file.
		The second alternative for the second
24		secure container is the signed .NET
25	<u> </u>	assembly. The second rule set is the
23		declarative statements in the .NET
26		assembly's header.
	means for creating and storing a third	When the second secure container is the
27	secure container; and	signed msi file, the third secure container is the signed .NET assembly.
28		die signed .ivi. assembly.
20		When the second secure container is the
Î		li II

Exhibit B

293482.02

1

	1	
1		
		signed .NET assembly, the third secure
2		container a .NET component secured by
	.	placing it in a signed cabinet file or by
3		including its hash in the header of the
		assembly.
. 4	means for copying or transferring at least a	When the second secure container is the
•	portion of said first protected information	signed msi file and the third secure
5	and a third rule set governing use of said	container is the signed .NET assembly, the
	portion of said first protected information	third rule set is the set of declarative
6	to said second secure container, said means for copying or transferring comprising:	statements within the assembly's header.
	for copying of transferring comprising.	When the second secure container is the
7	·	signed .NET assembly, the third rule set is
0		the set of declarative statements comprising
8		the LicenseProviderAttribute (third rule
. 9		set) added to the .NET component by its
、 ブ		developer.
10	means for incorporating said third	When the second secure container is the
10	secure container within said second	signed msi file and the third secure
11	secure container.	container is the signed .NET assembly, the
		assembly is placed in the signed .msi file.
12		3371 43 44 44 44 44 44 44 44 44 44 44 44 44
		When the second secure container is the
13		signed NET assembly and the third secure
		container is a .NET component contained in a signed cabinet file or a .NET
14		component whose hash is included in the
15		header of the assembly, the third secure
12		container is incorporated within the .NET
16		assembly.
10		
17	82. A data processing arrangement as in	
•	claim 81 further comprising:	
18	means for applying at least one rule from	When the third rule set is declarative
	said third rule set to at least in part govern	statements within the assembly's header, it
. 19	at least one factor related to use of said	governs the use of the .NET assembly
٥à	portion of said first protected information.	which includes the first protected information.
20		inomation.
21		When the third rule set is the set of
41		declarative statements comprising the
22		LicenseProviderAttribute added by the
22		NET component by its developer, it
23		ensures the user is licensed.
24	83. A data processing arrangement as in	
	claim 82 further comprising:	
25	means for applying at least one rule from	When the second rule set is the conditional
- 1	said second rule set to at least in part	syntax statements written by the .NET
26	govern at least one factor related to use of	assembly developer and placed into the
į	said portion of said first protected	signed .msi file, it governs the
27	information.	offer/installation of the .NET component.
	İ	
28	•	When the second rule set is the declarative
ļ.	<u> </u>	statements in the .NET assembly's header,
- 1		

Exhibit B

293482.02

### INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP. INTERTRUST INFRINGEMENT CHART

	FOR U.S. P.	ATENT NO. 5,915,019
5	85. A method comprising the following steps:	Infringing products include the .NET Framework SDK, Microsoft Visual Studio .NET, the Microsoft Installer SDK, and products that include the Microsoft .NET CLR, and the Microsoft Installer technology.
7	creating a first secure container comprising	The first protected information is the .NET component.
9		The first secure container is a signed .NET component (first protected information) governed by the set of declarative statements comprising the
10		LicenseProviderAttribute (first rule set).
11 12		The second alternative for the first secure container is a cabinet file signed by the .NET component developer containing a
13		(signed or unsigned) .NET component with license support. The first rule set is the set
14	storing said first secure container in a first	of declarative statements comprising the LicenseProviderAttribute.
15	memory;	The first secure container is stored at the .NET component developer's location.
16	creating a second secure container comprising a second rule set;	The first alternative for the second secure container is a signed .NET assembly and the second rule set is declarative
17		statement(s) within the assembly's header.
18		The second alternative for the second secure container is the signed .msi file in
20		which the .NET assembly developer packages its (signed or unsigned) assembly. The second rule set is the
21		conditional syntax statement(s) written by the .NET assembly developer and placed into the signed .msi file.
22	storing said second secure container in a second memory;	The second secure container is stored at the NET assembly developer's location.
23	copying or transferring at least a first	The .NET component developer packages
24	portion of said first protected information to said second secure container, said	its module in a signed .msi file for distribution to the .NET assembly
25	copying or transferring step comprising: creating a third secure container	developer's site.  The third secure container is the signed
26	comprising a third rule set;	.msi file in which the .NET component developer packaged its .NET component.
27	٠.	The third control set is the conditional syntax statements written by the .NET
28		component developer and placed into the signed .msi file.
	copying said first portion of said	In preparation for using a msi authoring

Exhibit B

•	.	
•	first protected information;	tool, such as Microsoft's Orca, copying the
2	2	NET component to a package staging area.
_	transferring said copied first portion	
. 3	of said first protected information to said third secure container; and	.NET component into the signed .msi file.
_		The NET example developed to 11 11
4	first portion of said first protected	The .NET assembly developer installs the .NET component, which involves
5		removing it from the .NET component
	container to said second secure	developer's signed .msi file and installing it
.6	container.	into its environment. Subsequently, the
		.NET assembly developer places the .NET
· 7	1	component into its .NET assembly and/or
8	·	signed msi file when it is packaging its.  NET assembly.
•	:	1 .IVET assembly.
9	87. A method as in claim 85 in which said	The entire .NET component is copied.
	copied first portion of said first protected	
10	information consists of the entirety of said	
11	first protected information.	
. 11	89. A method as in claim 85 in which	
12	said first memory is located at a first site,	The first memory is located at the .NET
		component developer's site.
13	said second memory is located at a second	The second memory is located at the .NET
14	site remote from said first site, and said step of copying or transferring said	assembly developer's site.
•	first portion of said first protected	The .NET component developer's signed .msi file is transferred from its site to the
15	information to said second secure container	site of the .NET assembly developer.
	further comprises copying or transferring	
16	said third secure container from said first	·
17	site to said second site.	
•	94. A method as in claim 85 further	
18	comprising:	
10	creating a fourth rule set.	When the second secure container is not a
19		signed .NET assembly, the fourth rule set is
20		declarative statements within the
~~		assembly's header.
21	1	When the second secure container is not
		the signed .msi file in which the .NET
22		assembly developer packages its (signed or
23		unsigned) assembly, the fourth rule set is
دے		the conditional syntax statements written
24		by the .NET assembly developer and placed into the signed .msi file.
		praced into the Signed Hist Hig.
25		
26		
26	•	
- 11		

## INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP. INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 5,915,019

. 3	1 OK 0.5.1 ATENT 110. 5,715,017	
. 3	85 (alternate infringing scenario)	
4 5 6	A method comprising the following steps:	Infringing products include the .NET Framework SDK, Microsoft Visual Studio .NET, the Microsoft Installer SDK, and products that include the Microsoft .NET CLR, and the Microsoft Installer
٠ _		technology.
7 8	creating a first secure container comprising a first rule set and first protected information;	The first protected information is the .NET component.
•	mornation,	The first alternative for the first secure
9		container is the signed and licensed .NET component. The first rule set is the set of
10		declarative statements comprising the
11		LicenseProviderAttribute in the .NET component.
12		The second alternative for the first secure
		container is a (signed or unsigned) .NET
13		component with license support contained within a cabinet file signed by the .NET
14		component developer. The first rule set is the set of declarative statements comprising
15		the LicenseProviderAttribute in the .NET component.
16		•
17		The third alternative for the first secure
17		container is the signed .msi file in which the .NET component developer packaged
18		its assembly. The first rule set is the
19		conditional syntax statements written by the .NET component developer and placed
19		into the signed .msi file.
20	storing said first secure container in a first	The first secure container is stored at the
	memory;	NET component developer's location.
21	creating a second secure container comprising a second rule set;	The first alternative for the second secure container is a signed .NET assembly and
. 22	Tomprioning a booting rate bot,	the second rule set is declarative
• •		statement(s) within the assembly's header.
23		The second alternative for the second
24.		secure container is the signed .msi file in which the .NET assembly developer
25		packages its (signed or unsigned)
26		assembly. The second rule set is the conditional syntax statement(s) written by
20	·	the .NET assembly developer and placed
27	·	into the signed .msi file.
	storing said second secure container in a	The second secure container is stored at the
28	second memory; copying or transferring at least a first	.NET assembly developer's location.  The .NET assembly developer places the
	Copyring of transferring at least a litst	The INE I assembly developer places the

Exhibit B

. 1		
. '	portion of said first protected information	.NET component into the third secure
. 2	to said second secure container, said	container, which is either a signed cabinet
	copying or transferring step comprising:	file or a signed .NET assembly.
3	creating a third secure container	When the second secure container is the
	comprising a third rule set;	signed .msi file, the third secure container
4		is the signed NET assembly. The third
5		rule set is the declarative statement(s) in the .NET assembly's header.
J	' <b> </b>	die INDT assembly's header.
6		When the second secure container is either
·	·	a .NET assembly or the signed .msi file, the
. 7	· .	third secure container is a signed cabinet
		file in which the .NET assembly developer
8		placed licensed .NET component. The
٠ ^	<b>∦</b> ÷	third rule set is the set of declarative
9		statements comprising the LicenseProviderAttribute in the .NET
10		component.
10	copying said first portion of said	Copying the NET component to either the
11	first protected information;	.NET assembly or to the signed cabinet
•		file.
12	transferring said copied first portion	Transferring the .NET component to either
12	of said first protected information to said third secure container; and	the .NET assembly or the signed cabinet file.
13	copying or transferring said copied	When the second secure container is the
14	first portion of said first protected	signed .msi file and the third secure
	information from said third secure	container is the signed .NET assembly, the
15	container to said second secure	.NET assembly is placed into the signed
	container.	msi file.
1.6	·	When the second grown and its
17		When the second secure container is either the .NET assembly or the signed .msi file
		and the third secure container is the signed
18		cabinet file, the signed cabinet file is placed
		into either the .NET assembly or the signed
19		.msi file.
20	87. A method as in claim 85 in which said	The entire NET
20	copied first portion of said first protected	The entire .NET component is copied.
21	information consists of the entirety of said	
	first protected information.	
22		
	93. A method as in claim 85 in which	
23	said step of copying transferring said	When the third secure container is the
24	copied first portion of said first protected	signed .NET assembly, it is placed in the
24	information from said third secure container to said second secure container	signed .msi file.
25	further comprises storing said third secure	When the third secure contains in a six at
	container in said second secure container.	When the third secure container is a signed cabinet file, it can be placed in either the
26		.NET assembly and/or the signed .msi file.
27	94. A method as in claim 85 further	
<u>.</u> ∦	comprising:	
28	creating a fourth rule set.	When the second rule set is declarative
H		statement(s) within the assembly's header,
H	· · · · · · · · · · · · · · · · · · ·	3.43.4. 45.4

Exhibit B

1		the fourth rule set is the conditional syntax statement(s) written by the .NET assembly
. 3		developer and placed into the signed .msi file.
4		When the second rule set is the conditional syntax statement(s) written by the .NET
. 5		assembly developer and placed into the signed .msi file, the fourth rule set is
6		declarative statement(s) within the assembly's header or the set of declarative
7		statements comprising the LicenseProviderAttribute in the .NET
8		component.
9	95. A method as in claim 94 further comprising:	
10	using said fourth rule set to govern at least one aspect of use of said copied first	If the fourth rule set is the .NET assembly developer's declarative statement(s) within
11	portion of said first protected information.	the .NET assembly's header, it governs the use of the .NET component.
12		If the fourth rule set is the conditional
13		syntax statements of the .NET assembly developer's signed .msi file, it governs the
14	· · · · · · · · · · · · · · · · · · ·	offer/installation of the .NET component.
15		•
. 16		•
17.		
18	·	
19		·
20		
21		
22	•	
23		
24		
25		
26	·	
27		
28		
.		i

# INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 5.915.019

•	3 FOR U.S.	PATENT NO. 5,915,019
;	85 (second alternate scenario for .NET	Framework SDK, Microsoft Visual Studien NET, the Microsoft Installer SDK, and products that include the Microsoft NET CLR, and the Microsoft Installer
	A method comprising the following grand	technology.
8	a first rule set and first protected	The first protected information is a .NET component.
. 9		The first alternative for the first secure
10		container is the signed and licensed .NET component. The first rule set is the set of
11		declarative statements comprising the LicenseProviderAttribute in the .NET component.
. 12		
13		The second alternative for the first secure container is a (signed or unsigned) .NET component with license support contained within a cabinet file signal.
. 14 15		within a cabinet file signed by the .NET assembly developer. The first rule set is the set of declarative statements comprising
16		the LicenseProviderAttribute in the .NET component.
17	·	The third alternative for the first secure
18		container is a .NET component whose hash is included in the assembly header of a .NET assembly. The first rule set is the set
19		of declarative statements comprising the LicenseProviderAttribute in the .NET component.
20	storing said first secure container in a first	
Z1	incinoi y.	The first secure container is stored at the .NET assembly developer's location.
22	creating a second secure container comprising a second rule set;	The second secure container is the -i-
23		developer packages its signed assembly
24		syntax statement(s) written by the NET
<u> </u>		assembly developer and placed into the signed msi file.
1 2	toring said second secure container in a econd memory;	The second secure container is stored at the
26    c	opying or transferring at least a firm	The .NET assembly developer places the
27   1c	said second secure container, said	ITE I CONTROLL INTO the third course
28 CI	eating a third secure container	assembly.
co	mpreno o think it	The third secure container is a signed .NET assembly and the third rule set is
-	Ex	hibit B

93482.02

	1	
	2	declarative statement(s) within the assembly's header.
	copying said first portion of said first protected information;	Copying the .NET component to the .NET assembly.
	transferring said copied first portion of said first protected information to	Transferring the .NET component to the
	said third secure container; and	
	first portion of said first protected	When the second secure container is the signed msi file and the third secure
7	container to said second secure	container is the signed .NET assembly, the .NET assembly is placed into the signed .msi file.
8	87. A method as in claim 85 in which said	
. 9	copied first portion of said first protected	The entire .NET component is copied.
10	first protected information.	
	90. A method as in claim 85 in which	
11	said first memory and said second memory are located at the same site.	First and second memory is at the .NET assembly developer's location.
12		assembly developer's location.
13	93. A method as in claim 85 in which	
1.4	said step of copying transferring said	When the third secure container is the
14	copied first portion of said first protected	signed .NET assembly, it is placed in the
15	information from said third secure container to said second secure container	signed .NET assembly, it is placed in the signed .msi file.
		signed .NET assembly, it is placed in the
15	information from said third secure container to said second secure container further comprises storing said third secure	signed .NET assembly, it is placed in the
15 16	information from said third secure container to said second secure container further comprises storing said third secure	signed .NET assembly, it is placed in the
15 16 17	information from said third secure container to said second secure container further comprises storing said third secure	signed .NET assembly, it is placed in the
15 16 17 18	information from said third secure container to said second secure container further comprises storing said third secure	signed .NET assembly, it is placed in the
15 16 17 18	information from said third secure container to said second secure container further comprises storing said third secure	signed .NET assembly, it is placed in the
15 16 17 18 19 20	information from said third secure container to said second secure container further comprises storing said third secure	signed .NET assembly, it is placed in the
15 16 17 18 19 20 21	information from said third secure container to said second secure container further comprises storing said third secure	signed .NET assembly, it is placed in the
15 16 17 18 19 20 21 22	information from said third secure container to said second secure container further comprises storing said third secure	signed .NET assembly, it is placed in the
15 16 17 18 19 20 21 22 23	information from said third secure container to said second secure container further comprises storing said third secure	signed .NET assembly, it is placed in the
15 16 17 18 19 20 21 22 23 24	information from said third secure container to said second secure container further comprises storing said third secure	signed .NET assembly, it is placed in the
15 16 17 18 19 20 21 22 23 24 25	information from said third secure container to said second secure container further comprises storing said third secure	signed .NET assembly, it is placed in the

## INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP. INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 5,915,019

3		· · · · · · · · · · · · · · · · · · ·
	96. A method comprising performing the	A signed and licensed .NET component .
.4.	following steps within a virtual distribution	(first container) is part of a .NET assembly
-	environment comprising one or more	(second container), which is packaged in a
5	electronic appliances and a first secure	signed .msi file (third container).
_	container, said first secure container	
6	comprising a first control set and first protected information:	
7	using at least one control from said first	The first secure container is a licensed and
	control set to govern at least one aspect of	signed .NET component governed by the
8	use of said first protected information	set of declarative statements comprising the
	while said first protected information is	LicenseProviderAttribute (one control).
9	contained within said first secure container;	
10	creating a second secure container	The second secure container is a .NET
10	comprising a second control set for governing at least one aspect of use of	assembly, the protected information is the
11	protected information contained within said	assembly and the second control set is
11	second secure container;	declarative statement(s) within the assembly's header.
12	incorporating a first portion of said first	Included in the .NET assembly is the .NET
	protected information in said second secure	component.
13	container, said first portion made up of	
	some or all of said first protected	
14	information;	
	using at least one control to govern at least	The declarative statement(s) govern the use
15	one aspect of use of said first portion of	of the .NET component and the custom
16	said first protected information while said	LicenseProvider class (first control set)
10	first portion is contained within said second secure container; and	controls the .NET component.
17	incorporating said second secure container	The third course contained is the single
_ ^ ′ .	containing said first portion of said first	The third secure container is the signed .msi file in which the .NET assembly
18	protected information within a third secure	developer packages its assembly. The third
	container comprising a third control set.	control set is the conditional syntax.
19	,	statements written by the assembly
		developer and placed into the signed .msi
20		file.
	· ·	
21		

_	INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.	
2	INTERTRUST INFRINGEMENT CHART	
3	FOR U.S. PATENT NO. 5,949,876	
4.		
·5	2.	Infringement is based on Microsoft's Visual Studio .NET and/or the .NET Framework licensing tools (in
6		the.NET Framework SDK) and/or Microsoft Installer SDK
7	A system for supporting electronic commerce including:	
8	means for creating a first secure control set at a first location;	The first location is a .NET component developer's site.
9		The first secure control set is the set of declarative statements comprising the License Provider Attribute of
10		a first .NET licensed component that provides for a design-time license to use the control. This attribute
11		also specifies the type of license validation that occurs. The component is encapsulated in a signed .NET
12	means for creating a second secure	assembly. The second location is the .NET application
13	control set at a second location;	developer's site where a .NET application comprising one or more assemblies is created.
14		
15		The second secure control set comprises the declarative statement(s) (including licensing statements, and code access security statements) of a
16		signed .NET assembly using or calling the first .NET component. The control set can include a set of
17		security permissions demanded by the .NET assembly containing the licensed component, whereby the
18		permissions are demanded of components that call the application components. The control set can also be
19		extended by controls expressed as conditional syntax statements in a signed .msi file containing a click
20		through end-user license (the end-user license scenario).
21	means for securely communicating said first secure control set from said first	The first .NET control set is securely communicated from the first location developer to the .NET solution
22	location to said second location; and	provider by either being contained in a signed assembly, within a signed cabinet file or within a
23	means at said second location for	signed .msi file.  At the second location, the solution developer uses the
24	securely integrating said first and second control sets to produce at least a	.NET runtime that includes the LicenseManager.
25	third control set comprising plural elements together comprising an	Whenever a class (control or component) is instantiated (here, an instance of the first .NET
26	electronic value chain extended agreement.	licensed component), the license manager accesses the proper validation mechanism for the control or
27	ug. comem.	component. A value chain is created through the creation of a run-time license for use of the first .NET
28		component in the context of use of the .NET
		application developed at the second location. The

Exhibit B

293482.02

. 1		
3		license controls for the runtime license (derived from the design time license) are bound into the header of the .NET application assembly, along with the second control set.
4 5 6 7		The creation of runtime license controls is securely handled by Visual Studio.NET or the LC tool. Runtime licenses are embedded into (and bound to) the executing assembly. The license control attribute included in the first .NET component is customized in the second location to express and require the runtime license. In a different scenario, the LC tool is used to create a ".licenses file" containing licenses for multiple components, including runtime licenses for
8 9 10		components and classes created by the license provider. This .licenses file is embedded into the assembly.  The third control set is an extended value chain
11		agreement that comprises the runtime license controls for the first .NET licensed class (that had been bound to the assembly), the declarative controls provided by
12 13		the solution provider in the solution provider's assembly, and any runtime licenses for other components included by the solution provider in the
14 15		solution provider's assembly, and any end user license agreement provided by the application provider. The controls are typically integrated into the header of the .NET application assembly calling the first .NET licensed component.
16 17 18		A further "end user licensing scenario" occurs when, at the second location, the application developer packages the application into a signed .msi file that includes conditional syntax statement controls that
19 20		require that a user read and agree to an end user license agreement for the application and the embedded first component. The third control set includes a plurality of elements that include the run-
21		time licenses mentioned above, security permissions controls, EULA controls (a fourth control set), all securely bound into the signed .msi file.
22		
24	11. A system as in claim 2 in which said first location and said second location are contained within a Virtual Distribution	The Microsoft .NET Framework provides a Virtual Distribution Environment. Here the nodes are the Common Language Runtime
25	Environment.	instances that interpret the controls contained within .NET assemblies (among other functions).
26 27		Total Junctions).
28	29. A system as in claim 2 in which said first secure control set includes required	The licensing control in the first control set specifies the method required to validate
		Exhibit B

293482.02

	•
terms.	the license.
32. A system as in claim 2 in which said	The security permissions demanded (as
second secure control set includes required	described above) are required terms for
terms.	execution of the application code element
	1 sweether of the application code clement
60. A system as in claim 2 in which said	In the coopering whose the small and
means for securely integrating said first and	In the scenario where the application
second control sets includes a fourth	1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
control set.	file, the secure integration of the first and
control set.	second control sets is enhanced by the
i	tamper protection afforded by the signed
	.msi file. In the end user license scenario.
	fourth control set consisting of conditional
	syntax statements is included in the .msi
	file.
• .	
130. A system as in claim 2 further	The third control set is executed under the
including means for executing said third	auspices of the CLR
control set within a protected processing	adspices of the CLR.
environment.	
environment.	<u> </u>
122 A mintom on in alain, 120: 1:1	Long 111
132. A system as in claim 130 in which	The third control set is executed at an end-
said protected processing environment is	user site within the CLR.
located at a location other than said second	
location.	
161. A system as in claim 2 in which said	In the end user license scenario, the third
third control set includes controls	control set includes a fourth control set that
containing human-language terms	requires that the human user agree with
corresponding to at least certain of the	license terms displayed to the user. These
machine-executable controls contained in	human readable terms are referenced in the
said third control set.	conditional syntax statement controls
•	contained in the signed .msi file.
162. A method as in claim 161 in which	The .msi file is a data descriptor data
said human-language terms are contained	structure.
in one or more data descriptor data	Su detaire.
structures.	
	<u> </u>
170 A mistom on in alain 2 in alain 1	77
170. A system as in claim 2 in which said	The creation of the first licensed
means for creating a first secure control set	component, including its licensed controls
ncludes a protected processing	is carried out under the auspices of the
nvironment.	CLR.
71. A system as in claim 2 in which said	The application design time environment
neans for creating a second secure control	and the creation of the .NET application is
et includes a protected processing	carried out under the auspices of the CLR.
nvironment.	out micor are anapiecs of the CLR
77 A system as in claim 2 in which and	The 6
72. A system as in claim 2 in which said	The means for integrating the runtime
neans at said second location for securely	license with the application controls is
ntegrating includes a protected processing	carried out under the auspices of the CLR.
nvironment.	
29. A system as in claim 2 in which said	VS.NET runs under Windows.
	::
•	H .
•	1
E.	vhihili B

· : .	1	•
	means for creating a first secure control set includes an operating system based on or	
	compatible with Microsoft Windows.	
,4	330. A system as in claim 2 in which said	VS.NET runs under Windows.
	set includes an operating system based on or compatible with Microsoft Windows.	
. 6	means at said second location for securely	VS.NET runs under Windows.
. 7	integrating said first and second control sets includes an operating system based on	
8	or compatible with Microsoft Windows.	
. 9	346. A system as in claim 2 further comprising means by which said third	The third control set in the scenario described in the claim map for claim 2
10	control set governs the execution of at least one load module.	governs a portable .NET executable
11		designed to be loaded into the CLR environment (a CLR host).
12	347. A system as in claim 2 farther comprising means by which said third	The third control set in the scenario
13	control set governs the execution of at least one method.	described in the claim map for claim 2 governs a .NET executable. This
14		executable contains one or more methods.
15	349. A system as in claim 2 further comprising means by which said third	The third control set in the scenario described in the claim map for claim 2
15 16	349. A system as in claim 2 further comprising means by which said third control set governs the execution of at least one procedure.	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This
16	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
16 17	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
16 17 18	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
16 17 18 19	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
16 17 18 19 20	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
16 17 18 19 20 21	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
16 17 18 19 20 21 22	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
16 17 18 19 20 21 22 23 24	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
16 17 18 19 20 21 22 23 24 25	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
16 17 18 19 20 21 22 23 24 25 26	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
16 17 18 19 20 21 22 23 24 25	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more

## INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP. INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 6,112,181

. 3		
4	CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
5	48.	Infringing products include Microsoft SMS (Systems Management Server) 2.0 and subsequent versions.
6	A method for narrowcasting selected digital information to specified	
7	recipients, including:	
8	a) at a receiving appliance, receiving selected digital information from a	The receiving appliance is the client (e.g., end user computer in an Enterprise setting)
9	sending appliance remote from the receiving appliance,	receiving digital information (packages and/or advertisement files) from the sending
10		appliance, the centralized SMS database via a Client Access Point and/or Distribution Point set up on a server.
11		set up on a server.
12	the receiving appliance having a secure node and being associated with a specified recipient;	The "node" is "secure" as a result of SMS security, as well as how it identifies and selects clients.
13	, and a position of the same o	The "specified recipient" is the result of the collection identifying a specific client that
14		meets the criteria for a package or advertisement.
15		
16	i) the digital information having been selected at least in part based on	The digital information is a software package.
17	the digital information's membership in a first class, wherein the first class	or advertisement. The "first class membership was determined in part using rights
18	membership was determined at least in part using rights management	management information" reads on creating software packages (or advertisements) based
19	information; and	on attributes of the software.
20	ii) the specified recipient having	The "specified recipient" is the client selected
21	been selected at least in part based on membership in a second class, wherein	to receive a package or advertisement. That recipient is chosen based on a collection rule,
22	the second class membership was determined at least in part on the basis	or on the recipient's possession of a license.
23	of information derived from the specified recipient's creation, use of, or	
24	interaction with rights management information; and	
25	b) the specified recipient using the receiving appliance to access the	The receiving appliance is the client computer. The SMS agents on the client computer
26	received selected digital information in accordance with rules and controls,	receive, evaluate and take the appropriate action based on rules and controls governing
27	associated with the selected digital information.	the package and/or advertisement (i.e. the selected digital information)
28	mormution.	sciecieu uigiiai injormaiion).
	the rules and controls being enforced	Rules and controls are enforced by Agents on
	The same of the sa	

Exhibit B

•		
1 2	by the receiving appliance secure node.	the client (the secure node)
3	59. The method of claim 48 wherein	Event information includes SMS event
4	said received selected digital information is at least in part event information.	information, including Scheduling Classes.
6	63. The method of claim 48 wherein said received selected digital information is at least in part executable software.	All SMS packages must include a minimum of one program.
8	70. The method of claim 48 wherein said rules and controls at least in part govern usage audit record creation.	A control governs whether a MIF (management information file) is sent back to the SMS db after installation is done to report on the success or failure of the installation.
10:	89. The method of claim 48 wherein said receiving appliance is a personal computer.	The primary purpose of SMS is to manage software on personal computers throughout the
2	computer.	Enterprise.
3		
4	•	•
5		
6	·	

## INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP. INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 6,112,181

3		
4	E E E E E E E E E E E E E E E E E E E	IN THE CHATMONING INTERING EMENT AND ADDRESS OF THE CONTROL OF THE
4	48.	Infringing products include Windows
5		Media Player and Windows Media Rights
,	·	Manager
6	A method for narrowcasting selected	This claim pertains to Windows Media
. •	digital information to specified recipients,	Player with Individualized DRM Client and
7	including:	Windows Media Rights Manager used in
•		the context of a narrowcast pay-per-view
8	:	(hear) media distribution service.,
		simulcast and/or subscription services.
.9		Participation of the second of
•	(a) at a receiving appliance, receiving	Receiving appliance is a user's PC with
-10	selected digital information from a sending	individualized DRM client (secure node).
	appliance remote from the receiving appliance, the receiving appliance having a	Specified recipient is a user using the specific individualized DRM client to
11	secure node and being associated with a	access and render narrowcast pay-per-view
10	specified recipient	media, simulcast and/or subscription
12	op	services for which the user acquires a
13		license.
13		
14		
• •	(i) the digital information having been	The digital information is media that is
15	selected at least in part based on the digital	narrowcast to licensed recipients. These
	information's membership in a first class,	narrowcast streams are licensed to users
16	wherein the first class membership was	who have acquired licenses and whose PCs
	determined at least in part using rights	(appliances) support WMPs that have individualized DRM clients. This attribute
17	management information; and	is included in the signed WMA file header
•		and is used in the process of acquiring
18		licenses for access to the media. Media that
19	·	are licensed to the recipient have their
17	·	licenses bound to the recipient's
20	•.	Individualization module.
. 20	(ii) the specified recipient having been	The recipient is selected for this content
21	selected at least in part based on	based on the fact that the recipient is a
	membership in a second class, wherein the	member of the class of recipients who have
22	second class membership was determined	a license for the narrowcast media and
	at least in part on the basis of information	whose devices support WMP and
23	derived from the specified recipient's	individualized DRM clients. The
٠.,	creation, use of, or interaction with rights	recipient's machine must indicate support
24	management information; and	for individualization in challenges that are
		sent as part of requests for media in this
25	(h) the energified recipient using the	narrowcast class.
	(b) the specified recipient using the	Recipient's machine uses WMP and the individualized DRM client to access the
26	receiving appliance to access the received	narrowcast media in accordance with all
ا م	selected digital information in accordance with rules and controls, associated with the	rules associated with the media and
:27	selected digital information, the rules and	contained in the media license – in
20	controls being enforced by the receiving	particular, requirements that
28	appliance secure node.	individualization be supported.
	application receipt those.	· · · · · · · · · · · · · · · · · · ·

Exhibit B

293482.02

	· ·
	Maria de Salva de la companio de Maria de Companio de Maria de Companio de Maria de Companio de Maria de Compa
61. The method of claim 48 wherein said	The digital information is Windows Media
received selected digital information is at least in part entertainment information.	which encodes audio/visual entertainment
reast in part entertainment information.	content.
62. The method of claim 61 wherein said	Reads.on narrowcast Windows Media File
entertainment information is at least in part	that are music or audio/visual.
music information.	
67. The method of claim 48 wherein said	The license contains a digital certificate.
rules and controls at least in part use digital	The DRM client uses the certificate in the
certificate information.	license to verify this signature and to verify
	that the header has not been tampered with
.72. The method of claim 48 wherein said	The signed header contains at least one
rules and controls in part specifying at least one clearinghouse acceptable to	URL that indicates to the Windows Media
rightsholders.	Rights Manager the license clearinghouse to be used in acquiring licenses.
75. The method of claim 72 wherein said at	This clearinghouse is a license
least one acceptable clearinghouse is a rights and permissions clearinghouse.	clearinghouse responsible for mapping rights and permissions onto requested
	content or narrowcasts and binding them to
	the requesting client environment or user of
	this environment.
89. The method of claim 48 wherein said	Windows Media Player and the
receiving appliance is a personal computer.	Individualized DRM client run on a
	personal computer.

Exhibit B

### INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.

INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 6,112,181

. 3	1 010 0.011112.111 1101 0,112,102	
4	91	Infringing products include Windows Media Player and Windows Media Rights Manager
5	A method for securely narrowcasting selected digital information to specified	This claim pertains to Windows Media Player with Individualized DRM Client and
6	recipients including:	Windows Media Rights Manager used in
7		the context of a narrowcast simulcast, pay- per-view (hear) media distribution service.
· <b>8</b>		and/or subscription services. The content is delivered in a Protected Windows Media
9.		File.
10	(a) receiving selected digital information in a secure container at a receiving appliance remote from a sending appliance, the	Narrowcast content is received in a Protected Windows Media File. Receiving appliance is user's PC with individualized
11	receiving appliance having a secure node, the receiving appliance being associated with a receiving entity	DRM client (secure node).
12	(i) the digital information having	The digital information is media that is
13	been selected at least in part based on the digital information's	narrowcast to licensed recipients (for example, a sold-out concert is narrowcast
14	membership in a first class,	on the Internet to "the class of" licensed (or ticketed) viewers).
15	(ii) the first class membership	These narrowcast streams are licensed to
•	having been determined at least in	users who have acquired licenses and
16	part using rights management	whose PCs (appliances) support WMPs
17	information	that have individualized DRM clients. This
17		attribute is included in the signed WMA file header and is used in the process of
18	•	acquiring licenses for access to the media.
10		Media that are licensed to the recipient
19		have their licenses bound to the recipient's
20	(h) the receiving entity having been	individualization module.
20	(b) the receiving entity having been selected at least in part based on said	The recipient is selected for this content based on the fact that the recipient is a
21	receiving entity's membership in a second	member of the class of recipients who has a
-	class,	license for the narrowcast media.
22	(i) the second class membership	The recipient class is determined by the
23	having been determined at least in part on the basis of information	license bound to the user's device that supports WMP and individualized DRM
	derived from the recipient entity's	clients. The recipient's machine must
24	creation, use of, or interaction with rights management information	indicate support for individualization in challenges that are sent as part of requests
25	116110 Hamagement Information	for media in this narrowcast class.
	(c) receiving at the receiving appliance	Receives a protected Windows Media File
26	rules and controls in a secure container,	
	(i) the rules and controls having	Receives a license that is bound to the file
27	been associated with the selected	as well as to the specific DRM client
ეი ∦	digital information; and	individualization information.
28	(d) using at the receiving appliance the	Recipient's machine uses WMP and the
H	selected digital information in accordance	individualized DRM client to access the

Exhibit B

. 1		
:	with the rules and controls,	narrowcast media in accordance with all
2		rules associated with the media and contained in the media license – in
3		particular, requirements that
•	(i) the rules and controls being	individualization be supported.
4	enforced by the receiving appliance	The WMP and DRM client enforce the rules embedded in the Protected Windows
5	secure node.	Media File License.
_	104. The method of claim 91 wherein said	77. 1: 41: 6
. 6	received selected digital information	The digital information is Windows Media, which encodes audio/visual entertainment
. 7	includes entertainment information.	content.
	109. The method of claim 91 wherein said	77 11
8	rules and controls at least in part use digital	The license contains a digital certificate. The DRM client uses the certificate in the
· 9	certificate information.	license to verify this signature and to verify
10		that the header has not been tampered with.
10	114. The method of claim 91 wherein said	The signed header contains at least one
11	rules and controls specify at least one	URL that indicates to the Windows Media
12	clearinghouse acceptable to rightsholders.	Rights Manager the license clearinghouse to be used in acquiring licenses.
		to be used in acquiring neclises.
13	117. The method of claim 114 wherein said at least one acceptable clearinghouse is a	This clearinghouse is a license
14	rights and permissions clearinghouse.	clearinghouse responsible for mapping rights and permissions onto requested
. 1 -		content or narrowcasts and binding them to
15		the requesting client environment or user of this environment.
16		
17	131. The method of claim 91 wherein said receiving appliance is a personal computer.	Windows Media Player and the
	receiving appliance is a personal computer.	individualized DRM client run on a personal computer.
18		
19		
20		•
20		
21	•	12
22		
_	•	

Exhibit B

.

## INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP. INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 6,389,402

3		
4	HERENCEANMITANGUAGE TO THE	CFAIM OF INFRINGEMENTS OF THE SECOND
5 6	1.	Products infringing: Microsoft Visual Studio NET, NET License Compiler, NET Framework SDK, and NET Common Language Runtime
7 8 9	A method including	A method for producing a third .NET component (application) that incorporates first and second .NET component whose distribution is license controlled.
10	creating a first secure container including a first governed item and having associated a first control;	The first secure container is a first signed .NET component that includes a license control. The governed item is the .NET component.
12 13 14 15		The first control is the set of declarative statements comprising the LicenseProviderAttribute of a first .NET licensed component that provides for a design-time license to use the control. This attribute also specifies the type of license validation that occurs.
16 17 18	creating a second secure container including a second governed item and having associated a second control;	The second secure container is the second signed .NET component that includes a license control. The governed item is the .NET component.
19 20 ′ 21 22		The second control is the set of declarative statements comprising the LicenseProviderAttribute of a second .NET licensed component that provides for a design-time license to use the control. This attribute also specifies the type of license validation that occurs.
23	transferring the first secure container from a first location to a second location;	The creator distributes a signed and licensed .NET component.
24 25		An application developer at a second location downloads a first .NET component for inclusion into an application.
26 27 28	transferring the second secure container from a third location to the second location;	A creator distributes a signed and licensed .NET component from a different location.  Application developer downloads a second .NET component for inclusion into an
	Fxhi	application.

Exhibit B

293482.02

•		./
1		
2	at the second location, obtaining access to at least a portion of the first governed item, the access being governed at least in part by the first control;	At the second location, the application developer uses the .NET runtime that includes the LicenseManager to access a first governed item.
4 · 5 · 6 · 7		Whenever a class (control or component) is instantiated (here, an instance of the first .NET licensed component), the license manager accesses the proper validation mechanism for the control or component.
8 9		The first control comprises the declarative statement(s) (including licensing statements, and code access security statements) of the first .NET component.
10 11	at the second location, obtaining access to at least a portion of the second governed item, the access being governed at least in part by the second control;	At the second location, the application developer uses the .NET runtime that includes the LicenseManager to access a second governed item.
12	second condor,	Whenever a class (control or component) is
13		instantiated (here, an instance of the second .NET licensed component), the license manager accesses the proper validation
14		mechanism for the control or component.  The second control comprises the declarative
15		statement(s) (including licensing statements, and code access security statements) of the second .NET component.
16	at the second location, creating a third secure	At the second location, the application
17	container including at least a portion of the first governed item and at least a portion of the	developer uses the .NET runtime that includes the LicenseManager to access a first governed
18	second governed item and having associated at	item and second governed item to construct an application, the third secure container.
19	least one control, the creation being governed at least in part by the first control and the	
20	second control.	Creation governance is accomplished by invoking the .NET runtime to access the first governed item and the second governed item.
21		Whenever a class (control or component) is
22		instantiated the license manager accesses the proper validation mechanism for the control or
23		component.
24		The portions of the first governed item and second governed item that are being included
25		in the third secure container will typically include the governed items themselves, ie. the
26		NET components.
27		The associated control in this case is the LicenseProviderAttribute, created and inserted
28		into the application.